THE USE OF ARTIFICIAL INTELLIGENCE TECHNIQUES FOR POWER SYSTEM ANALYSIS

by

Bryan James Gwyn

This thesis is submitted for the degree of

Doctor of Philosophy

at

Department of Electrical, Electronics and Information Engineering

City University

London

UK

December 1999

CITY UNIVERSITY LONDON

ABSTRACT

This thesis reports the research carried out into the use of Artificial Intelligence techniques for Power System Analysis. A number of aspects of Power System analysis and its management are investigated and the application of Artificial Intelligence techniques is researched.

The use of software tools for checking the application of power system protection systems particularly for complex circuit arrangements was investigated. It is shown that the software provides a more accurate and efficient way of carrying out these investigations. The National Grid Company's (plc, UK) use of software tools for checking the application of protection systems is described, particularly for complex circuit arrangements such as multi-terminal circuits and composite overhead line and cable circuits. Also described, is how investigations have been made into an actual system fault that resulted in a failure of protection to operate. Techniques using digital fault records to replay a fault into a static model of protection are used in the example. The need for dynamic modelling of protection is also discussed. Work done on automating the analysis of digital fault records using computational techniques is described. An explanation is given on how a rule-based system has been developed to classify fault types and analyse the response of protection during a power system fault or disturbance in order to determine correct or incorrect operation.

The development of expert systems for on-line application in Energy Control Centres (ECC), is reported. The development of expert systems is a continuous process as new

knowledge is gained in the field of artificial intelligence and new expert system development tools are built. Efforts are being made for on-line application of expert systems in ECC as preventive control under normal/alert conditions and as a corrective control during a disturbance. This will enable a more secure power system operation. Considerable scope exists in the development of expert systems and their application to power system operation and control.

An overview of the many different types of Neural Network has been carried out explaining terminology and methodology along with a number of techniques used for their implementation. Although the mathematical concepts are not new, many of them were recorded more than fifty years ago, the introduction of fast computers has enabled many of these concepts to be used for today's complex problems. The use of Genetic Algorithm based Artificial Neural Networks is demonstrated for Electrical Load Forecasting and the use of Self Organising Maps is explored for classifying Power System digital fault records.

The background of the optimisation process carried out in this thesis is given and an introduction to the method applied, in particular Evolutionary Programming and Genetic Algorithms. Possible solutions to optimisation problems were introduced to be either local or global minimum solutions with the latter being the desirable result. The evolutionary computation that has potential to produce a global solution to a problem due to the searching mechanisms that are inherent to the procedures is discussed. Various mechanisms may be introduced to the genetic algorithm routine which may eliminate the problems of premature convergence, thus enhancing the methods' chances of producing the best solution. The other, more traditional methods of optimisation described include Lagrange multipliers, Dynamic Programming, Local Search and Simulated annealing. Only the Dynamic Programming method guarantees a global optimum solution to an optimisation problem, however for complex problems, the method could take a vast amount of time to locate a solution due to the potential for combinatorial explosion since every possible solution is considered. The Lagrange multiplier method and the local search method are useful for quick location of a global minimum and are therefore useful when the topography of the optimisation problem is uni-modal. However in a complex multi-modal problem, a global solution is less likely. The simulated annealing method has been more popular for solving complex multi-modal problems since it includes techniques for the search to avoid being trapped in local minimum solutions.

Artificial Neural Network and Genetic Algorithm have been used to design a neural network for short-term load forecasting. The forecasting model has been used to produce a forecast of the load in the 24 hours of the forecast day concerned, using data provided by an Italian power company. The results obtained are promising. In this particular case, the comparison between the results from the Genetic Algorithm – Artificial Neural Network and Back Propagation - Neural Network shows that the Genetic Algorithm – Artificial Neural Network does not provide a faster solution than the Back Propagation - Neural Network.

The application of Evolutionary Programming to fault section estimation is investigated and a comparison made with a Genetic Algorithm approach. To enhance service reliability and to reduce power outage, rapid restoration of power system is required. As a first step of restoration, the fault section should be accurately estimated quickly. The

Fault Section Estimation (FSE) identifies fault components in a power system by using information on the operation of protection relays and circuit breakers. However this task is difficult especially for cases where the relay or circuit breaker fails to operate and for multiple faults. An Evolutionary Programming (EP) approach has been developed for solving the FSE problem including malfunctions of protection relays and/or circuit breakers and multiple fault cases. A comparison is made with the Genetic Algorithm (GA) approach at the same time. Two different population sizes are tested for each case. In general, EP showed faster computational speed than GA with an average factor of 13 times more. The final results were almost the same. The convergence speed (the required number of generations to get an optimum result) is a very important factor in real time applications. Test results show that EP is better than GA. However, as both EP and GA are evolutionary algorithms, their efficiencies are largely dependent on the complexity of the problem that might differ from case to case.

The use of Artificial Neural Networks to classify digital fault records is investigated showing theat Self Organising Maps could be useful for classifying records if integrated into other systems. Digital fault records are a very useful source of information to the protection engineer to assist with the investigation of a suspected unwanted operation or failure to operate of a protection scheme. After a widespread power system disturbance, due to a storm for example, a large number of fault records can be produced. A method of automatically classifying fault records would be very helpful in reducing the amount of time spent in manual analysis, thus assisting the engineer to focus on records that need in depth analysis. Fault classification using rule base methods have already been developed. The completed work is preliminary in nature and an overview of an extension to this work, involving the extraction of frequency components from the digital fault record data and using these as input to a SOM network, is described.

SUMMARY OF ORIGINAL CONTRIBUTIONS

- The use of Software tools for Investigating the Performance of Power System Protection
- Genetic Algorithm based Artificial Neural Networks for Electric Load Forecasting
- Application of Evolutionary Programming to Fault Estimation
- The use of Artificial Neural Networks to Classify Faults from Digital Fault Records

LIST OF PUBLICATIONS

L L Lai, A G Sichanie and B J Gwyn, 'A comparison between evolutionary programming and genetic algorithm for fault section estimation', IEE Proceedings, Generation, Transmission and Distribution, Vol 145, No 5, Sept 1998, 616-620

L L Lai, H Subasinghe, N Rakumar, E Vaseekar, B J Gwyn and V K Sood, 'Object-oriented genetic algorithm based artificial neural network for load forecasting', Lecture Notes in Computer Science, LNCS, Springer-Verlag, Xin Yao et al. (Editors), May 1999.

J A Downes, J L H Goody, B J Gwyn, 'The use of Current Software Tools for Investigating the Performance of Power System Protection in the National Grid Company' - CIGRE - study committee 34 colloquium and meeting, Stockholm 1995

ACKNOWLEDGEMENTS

I would like to thank my wife, Meryl for supporting me over the years especially when I was writing up this thesis. Many half terms and Saturdays I have been absent from home tapping away at this work – thanks Mel!
My two children, Sarah and Christopher and Grandpa Gwyn, thanks Dad!

I would like to thank Dr Lai for his inspiration and guidance in carrying out this research. I will always remember his immortal catch phrase " **Don't panic!**"

I would like to The National Grid Company for their support and for the opportunity to do this work.

The author wishes to acknowledge the research and development work carried out by The Centre for Electrical Power Engineering at the University of Strathclyde, on Dynamic Modelling of Power System Protection and The Department of Cybernetics at The University of Reading, on Automatic Fault Record Analysis. In addition, the development work of Power System Management Corporation, Pennsylvania, USA is duly acknowledged.

DECLARATION

The author hereby grants powers of discretion to the City University Librarian to allow this thesis to be copied in whole or in part without further reference to the author. This permission covers only single copies made for study purposes, subject to normal conditions of acknowledgement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter I

INTRODUCTION

Power Transmission Networks are highly complex Systems and play a vital role in modern society. In the UK, since the privitisation of the industry in 1990, there have been major changes in the way the Power System is operated. These changes have required engineers to provide smarter cheaper solution to more and more complex problems in shorter timescales. The use of new technology has played a vital role in enabling these solutions.

The use of software tools for checking the application of power system protection systems particularly for complex circuit arrangements, such as multi-terminal circuits and composite overhead line and cable circuits, are an example of the application of artificial intelligence techniques that have been applied in investigations into actual system faults, that resulted in a failure of power system protection to operate. A rule-based system has been developed to classify fault types and analyse the response of protection during a power system fault or disturbance in order to determine correct or incorrect operation. With the increasing demands made on the power system and its protection, the use of computer software tools for carrying out application and fault investigation studies is becoming more important.

It is important to plan and operate power systems such that there is enough redundancy in generation and transmission capacities to cope with the changes due to these disturbances and there are enough controllable resources (generation re-dispatch, load control, etc).

Such 'robustness' of the system relative to imminent disturbances is referred to as the security of the system. Security analysis and control have been implemented in modern Energy Control Centres (ECC), by software package called *Energy Management Systems* (EMS). EMS contains powerful Power Application Software (PAS) like state estimation, scenario analysis, operator load flow, short circuit analysis, stability analysis, optimal power flow with security constraints are major components of an EMS.

An estimate of the system state variables (bus voltage magnitudes and phase angles, for normal steady state) is obtained by *state estimation.* The line loading and other information concerning system operation may be derived from the estimated state variables. State estimation provides the real time database for other analysis, control and optimisation functions. To assess whether a normal operating state is secure or not, the system has to be subjected to contingency analysis. Contingency analysis is a software application run in EMS to give the operators an indication of what might happen to the power system in the event of a probable equipment outage.

In the late 1980's an area called Connectionism, Neural Networks or Parallel Distribution Processing emerged as a topic for research and for commercial development. Neural Networks were not entirely a new field [25, 26, 27], what was new was the concern with well-founded analysis and a deepening understanding of the topic. The topics mentioned all refer to machines that unlike conventional computers have a structure that reflects the structure of the brain. Biological computing, the brain and the nervous system of animals and human beings have existed for millions of years. They are effective in processing sensory information and controlling the interactions of animals within their environment.

One fascinating property of Artificial Neural Networks is their ability to exceed the limitations of traditional information processing such as the need for detailed programming. Neural Networks have many interesting properties such as their capacity for adaptation, use of distributed memory, capacity for generalization, ease of construction and their parallelism.

Although Artificial Neural Networks do have several fascinating properties they also have several limitations. Most networks are simulated on sequential machines, which means as the problem gets larger the processing time rapidly increases. The main disadvantage is their inability to explain any results obtained. The quality of their performance is measured by statistical methods.

Problems which are suitable for implementation by Artificial Neural Networks include pattern recognition, signal processing, Vision, speech processing, forecasting and modeling decision making aids and robotics. Artificial Neural Networks are used in industrial applications, the financial sector, the telecommunications sector and the environment sector.

Optimisation is the mathematical process of finding a better, or ideally the best or optimum strategy amongst multiple alternatives that will perform a certain task. Seeking such a strategy has in recent years become more viable for complex engineering systems with the advent of faster and more affordable computing power in the form of everyday personal computers. Of particular relevance to the topic of this documentation is the opportunity to save costs in the area of on-line engineering process control.

The genetic algorithm is a particular optimisation and search method that was first proposed by Holland in the 1970s and has proven to be a powerful and general technique [34]. The generality of the method stems from the methods' ability to perform well when little or no domain knowledge is available. It is therefore useful for solving complex engineering system optimisation problems and yet engineers and computer scientists are only just beginning to realise the benefits of such theory [35]. Evolutionary Computation [36] is a name that has been given to describe the field of research that investigates the solution of practical problems by the use of computational processes that simulate natural evolution and genetics.

An accurate and stable load forecast is essential for many operating decisions taken by utilities. The short-term load forecast provides the information to be adopted in the on-line scheduling and security functions of the energy management system, such as unit commitment, economic dispatch and load management. Hence, accurate load forecasting is essential for the optimal planning and operation of large-scale power systems.
Many techniques have been proposed and used for short-term load forecasting. Time-series models based on extrapolation are used for the representation of load behavior by trend curves. The time series approach, regression approach, state-space models, pattern recognition and expert systems are also some of the other techniques used [73-77]. All the above approaches use a large number of complex equations that involve lots of computational time. More recently, artificial neural network (ANN) techniques have been used in many modelling problems [78-82].

To enhance service reliability and to reduce power outage, rapid restoration of power systems is required. As a first step of restoration, the fault section should be accurately estimated quickly. The Fault Section Estimation (FSE) identifies fault components in a power system by using information on the operation of protection relays and circuit breakers. However this task is difficult especially for cases where the relay or circuit breaker fails to operate and for multiple faults.

Several papers have reported surveys on Evolutionary Algorithms (EA) applications in power systems [83,84]. Few methods have so far been employed to solve the FSE problem. This includes - Expert Systems [85] and other Computational Intelligence Techniques (CIT), such as, Artificial Neural Networks [86,87] and Genetic Algorithms [88]. Among these methods, expert systems are based on the production of a great number of rules describing the complex protection system behaviour. This will result in the maintenance of such a complex rule based knowledge base is very difficult, costly and time consuming. The response time for complex systems is also long.

Digital fault records are a very useful source of information to the protection engineer to assist with the investigation of a suspected unwanted operation or failure to operate of a protection scheme. After a wide spread power system disturbance due to a storm for example, a large number of fault records can be produced. A method of automatically classifying fault records would be very helpful in reducing the amount of time spent in manual analysis, thus assisting the engineer to focus on records that need in depth analysis. Fault classification using rule base methods have already been developed.

A noticeable advantage of an ANN over rule based systems is that they are capable of dealing with input vectors of data that are partially incomplete or incorrect. A trained ANN is capable of doing this because what it learns about one pattern generalises to other similar patterns. Conflicting information does not paralyse the ANN, it will still make its best judgement based on the information available. On the other hand, the rule-based system will typically fail if not presented with complete and accurate input data. An ANN could therefore compliment a rule based system for classifying digital fault records, in offering the protection engineer a possible solution when the rule based system fails to recognise a particular fault.

This thesis explores the techniques highlighted above in various areas of Power System Management. Each chapter investigates a particular aspect of Power System management and the application of a particular Artificial Intelligence technique.

Chapter II

THE USE OF CURRENT SOFTWARE TOOLS FOR INVESTIGATING THE PERFORMANCE OF POWER SYSTEM PROTECTION IN THE NATIONAL GRID COMPANY PLC, UK

2.1 Introduction

This Chapter describes The National Grid Company's (plc, UK) use of software tools for checking the application of protection systems, particularly for complex circuit arrangements such as multi-terminal circuits and composite overhead line and cable circuits. This chapter also describes how investigations have been made into an actual system fault that resulted in a failure of protection to operate. Techniques using digital fault records to replay a fault into a static model of protection are used in the example. The need for dynamic modelling of protection is also discussed. Finally, work done on automating the analysis of digital fault records using computational techniques is described. This part will explain how a rule-based system has been developed to classify fault types and analyse the response of protection during a power system fault or disturbance in order to determine correct or incorrect operation.

With the increasing demands made on the power system and its protection, the use of computer software tools for carrying out application and fault investigation studies is becoming more important. In order to meet these needs a suite of software tools has been introduced and more are in the course of investigation and development.

2.2 Application Studies

A particular software tool, FR10 [2] designed for protection application studies consists of a network model and appropriate relay models. The network model consists of an interconnected network of seven circuits consisting of five 2-terminal circuits and two 3-terminal circuits as shown in figure 2.1. There are four possible sources of' power in-feed into the network which can be controlled by simulated circuit breakers at

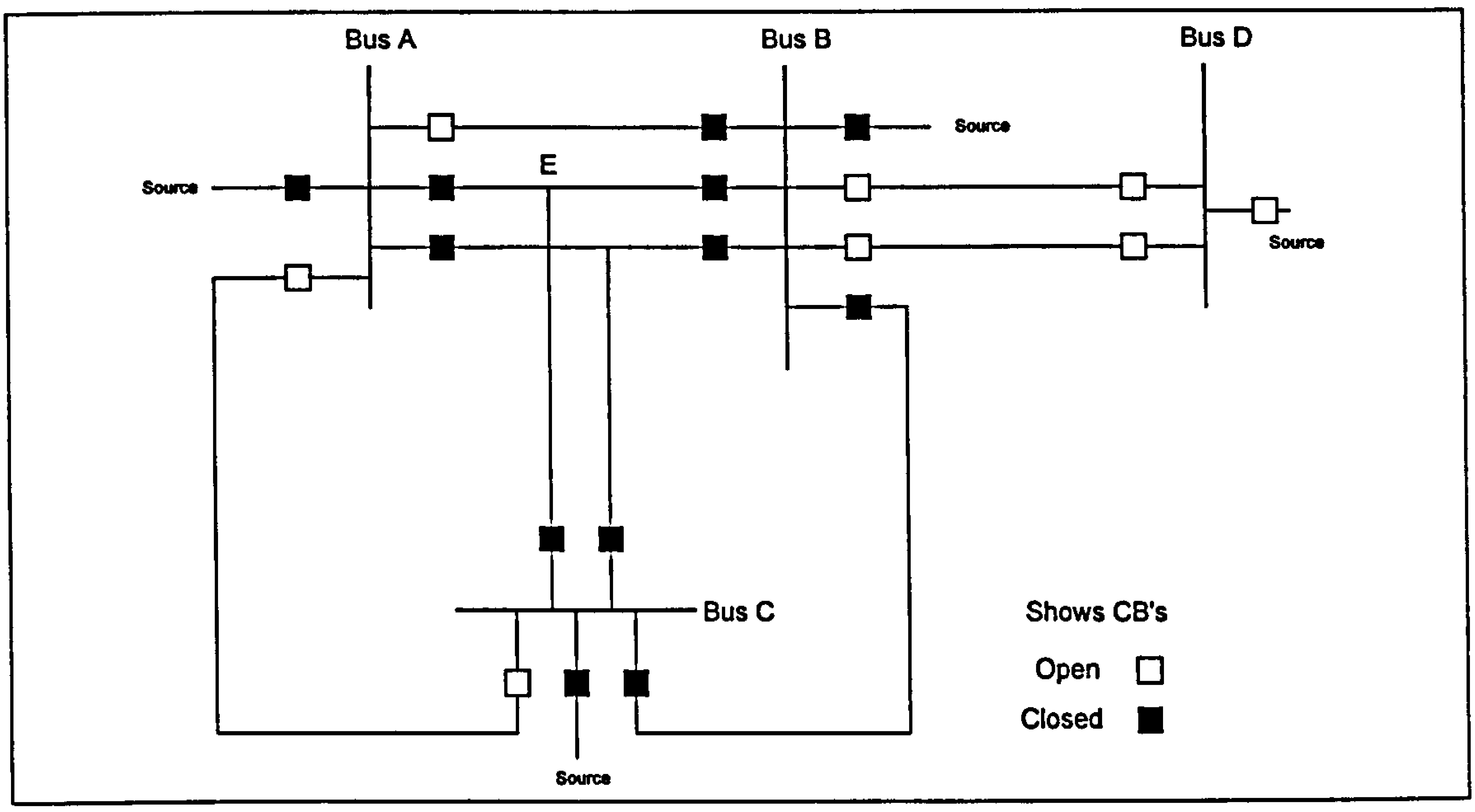

Figure 2.1: Circuit Configuration showing impedances on reduced model

each node. Values of sequence impedances of the lines and in-feeds can be entered into the model. The location of the fault, and its type, with a range of fault resistances can also be entered. The relay model consists of comparator equations which can be set by assigning values to constants which describe the relay characteristics.

Figure 2.2 shows a plot of the zone 1 characteristics of a distance relay for an under-reaching scheme on the BE leg at bus B of the teed circuit configuration in figure 2.1. In this configuration leg AE was 19Ω compared with 4Ω for leg BE and 3Ω for leg CE. The characteristic shown in this case is of a combination of mho and quadrilateral types for earth faults. In figure 2.2 an earth fault at the tee point E with fault resistance values between zero and 5Ω is Plotted.

When the response of the same distance relay on leg AE at end A was studied, it was found that the reach of zone 1 could not be set to guarantee detection of faults up to the tee point. It followed that, even with direct inter-tripping facilities, fast fault clearance

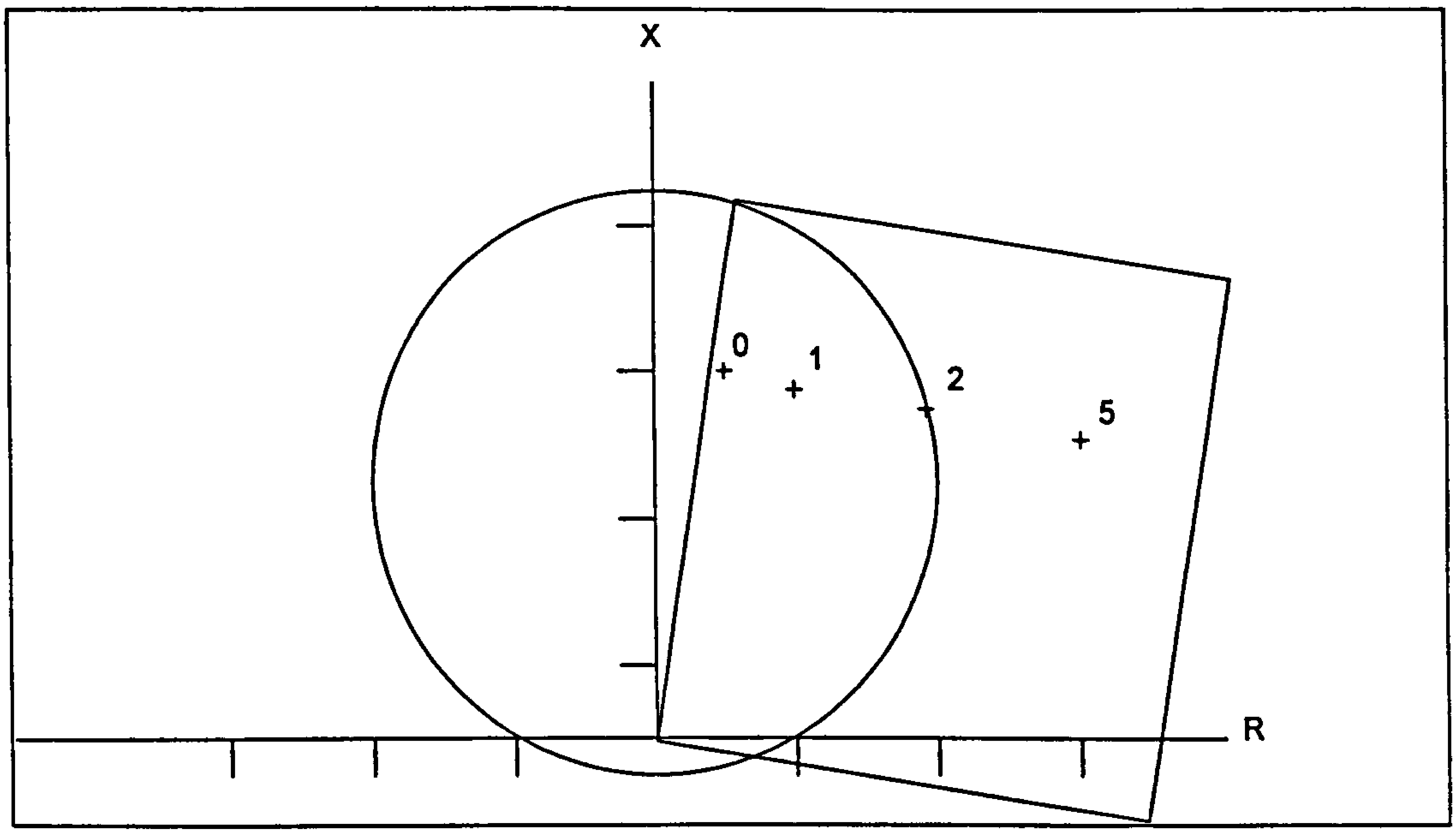

Figure 2.2: Characteristics of distance relay at bus B to E (values are in Ohms)

time could not be guaranteed for all faults occurring within the circuit using a zone 1 under-reaching distance protection. However, further study indicated that an additional instantaneous zone 2 blocked over-reach facility could be used to overcome such a limitation.

2.3 Fault Investigations

Following incidents or faults, the first detailed source of information is commonly provided by the paper outputs of digital fault recorders. These paper records give a graphical representation of primary system voltage and current together with the response of protection relay outputs. A brief visual inspection of these paper records is usually all that is needed for a protection engineer to obtain a satisfactory assessment of the correctness or otherwise, of the protection operations. However, in some instances a deeper level of analysis is required. A software tool [2] is available to provide this deeper level of analysis.

In the same manner as described in Section 2.2, the main feature of the software tool is its ability to model the characteristics of a range of protection relay types and settings, and the impedance parameters of associated primary circuits. Instead of having to set up various theoretical scenarios on the primary system, this software can replay the actual primary system conditions into the relay model. This is done by extracting the digital information held in the memory of the fault recorders and then loading it into the database of the software. The software is designed to read the fault records of a number of different manufacturers fault recorders and converts them to a common format which can be displayed with cursor control for manual analysis. Other useful features of this software include harmonic analysis, identification of faulted phases, and distance-to-fault estimation using information from all available ends of the circuit.

The relay evaluation software is designed to use the fault record information from the two line ends in the case of a two-terminal circuit or three line ends in the case of a three-terminal circuit. In the instances when only one record associated with a particular fault is available, the software can still be used by estimating in-feeds from the remote end of the circuit in the manner described in Section 2.2

An example is now given of how the software provided an explanation, of the second main distance protection response at both ends of a line which failed to operate in zone 1, for a phase-to-earth fault on the line itself. For this fault the first main unit protection operated correctly.

Figure 2.3 shows how the model represents end A of a line protected by an offset mho relay with reach settings of 80%, 150% and 350% of the protected line for zones 1, 2 and 3 respectively. Figure 2.4 shows a fault recorder record of the primary current and voltage during the fault at end A which was classified by the

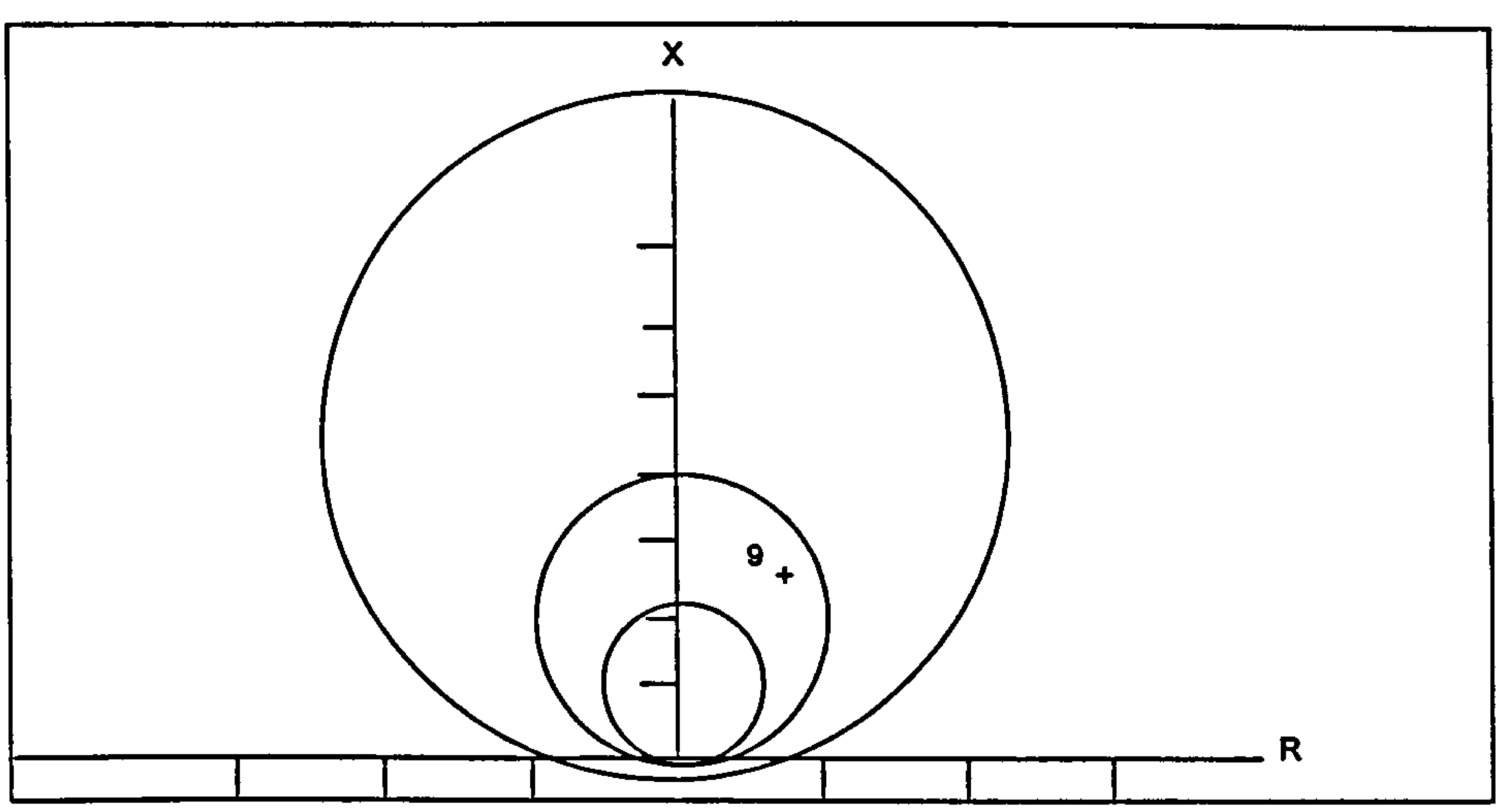

Figure 2.3: Example of zone 1 non-operation, End A; zones 1, 2 and 3

Software (as can be seen by inspection) as a c-phase to earth fault. The software estimated the distance-to-fault to be 85% of the line length from end A and the fault resistance was estimated to be 9Ω. The resistance value of 9Ω plotted in figure 2.3 for end A, and figure 2.5 for end B are outside the zone 1 characteristic but inside zone 2. This investigation showed that the resistance of the fault was too large to be detected in zone 1 at either end with the system conditions applying at the time.

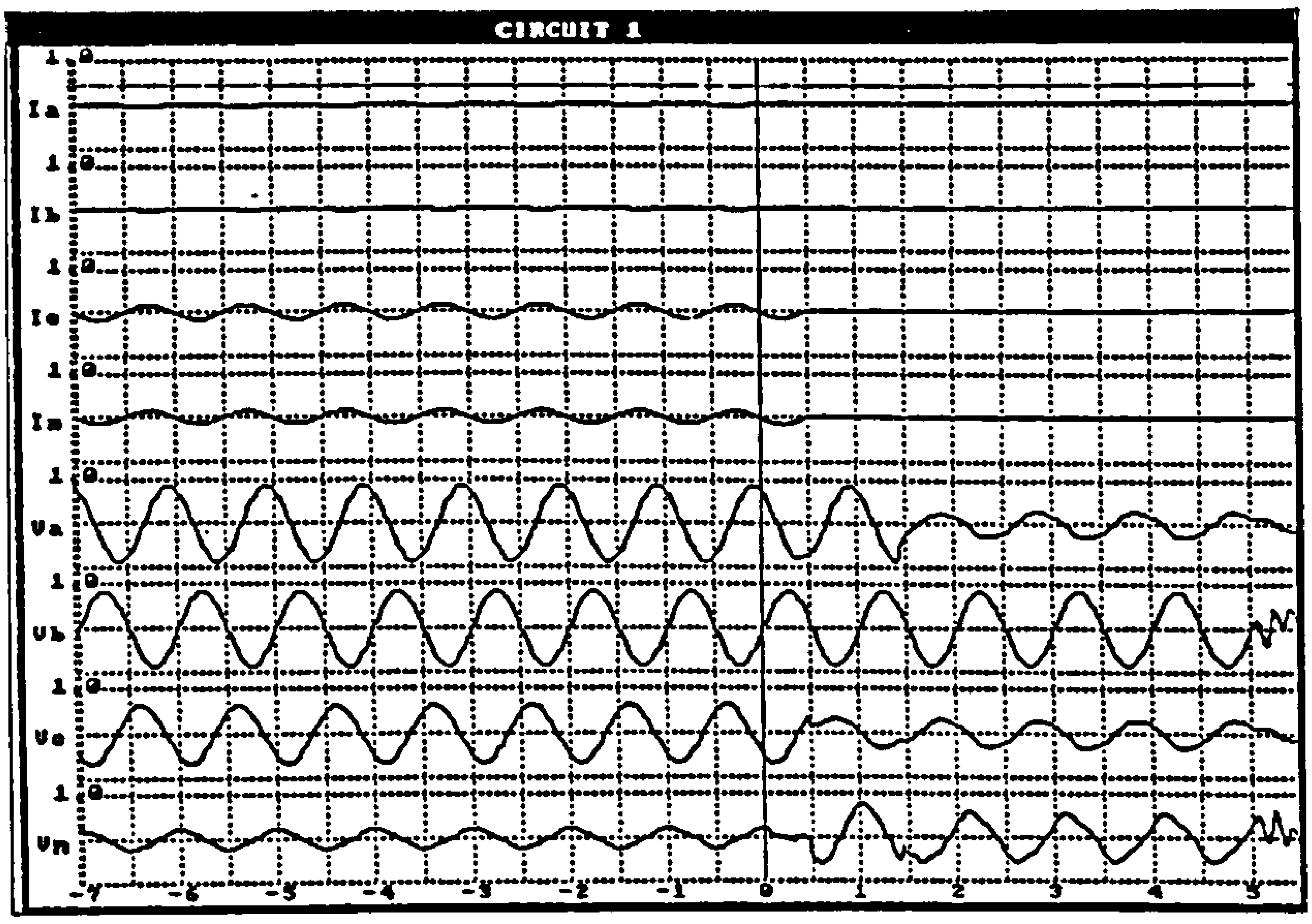

Figure 2.4: Voltage and current records for a c-phase to earth fault

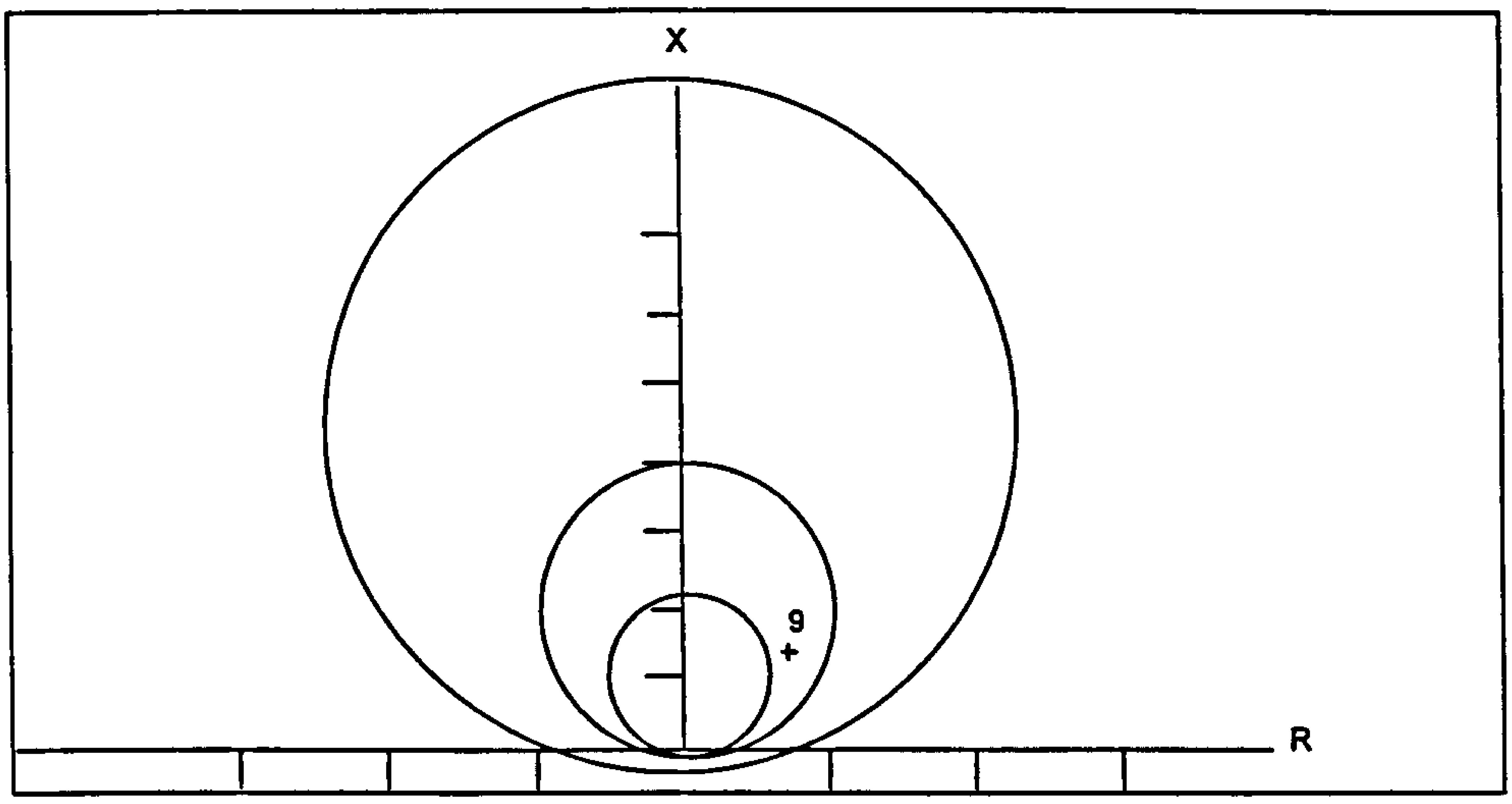

Figure 2.5: Example of zone 1 non-operation, End B; zones 1, 2 and 3

2.4 Dynamic Modelling

The software packages described in sections 2 and 3 model the static characteristics of protection relays. Moreover, steady-state sinusoidal conditions are assumed in the system model described in Section 1.2. Thus the possible effects of any transients or frequencies other than 50Hz are not taken into account in these models. Although protection is generally designed to minimise the effects of these phenomena, in some circumstances, the potential exists for protection to respond incorrectly due to such conditions. Thus, software that can model the dynamic response of the primary system, its transducers and connected protection will sometimes have an advantage over static models particularly when investigating some system incidents.

In order to provide this capability, an integrated dynamic model consisting of a power system module and a protection system module is being developed [3]. Referring to figure 2.6, the primary power system module provides the three-phase current and voltage which feeds a transmission line and primary transducers i.e. CTs and VTs, at a specified location. The protection system module will be made up of many protection model groups. Each digital protection model group will consist of a transducer, an analogue-to-digital converter, a relay and a circuit breaker model. The protection system module will process information received from the power system module, and output a status for the circuit breakers as time is incremented by the model controller. Thus the protection system module will form a closed feedback loop, which is illustrated in figure 2.6. The output of the software is in the form of a fault record, where all information from the modules is plotted against time.

A suite of relays models has been developed [4] including digital current differential, phase comparison unit protection and digital and analogue mho and quadrilateral distance protection. In one instance the relay algorithm was used in the model, A current differential protection (Microphase FM, Reyrolle Protection)

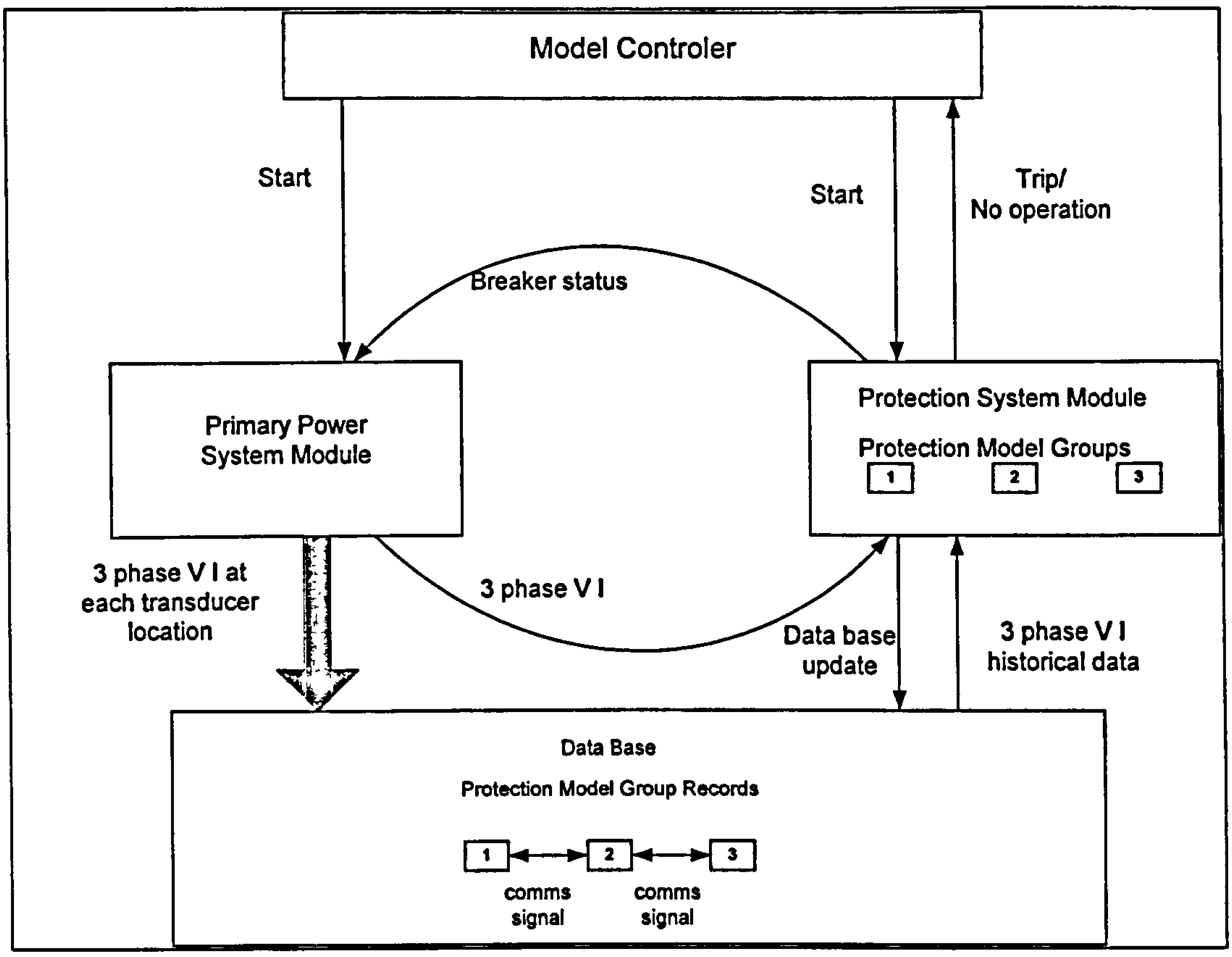

Figure 2.6: Structure of the co-ordinated dynamic power system model

2.5 Automatic Analysis of Fault Records

2.5.1 Automatic analysis of protection response

The fault recorders deployed throughout the National Grid system are triggered to provide records when circuit breakers trip and when disturbances cause protection relays to respond. Many fault records can be generated by one system fault or incident and particularly following multiple faults caused by adverse weather conditions. Under these conditions protection engineers have to deal with a large number of fault records. For each record the engineer needs to determine if the protection responded correctly and also to enter the data from the record into a fault analysis and defect system. This process is very time consuming. In order to reduce this time and to assist the engineer in focusing on records which need in-depth analysis as described in sections 2.2 and 2.3, software is being developed that will automate this analysis.

The software will analyse a batch of fault records extracted from a number of different types and manufactures of fault recorders [4]. The processes used by the software are as follows: -

i) It identifies the format of each record stored in the selected hard disk directory so that it can read the data. These formats can vary according to fault recorder manufacturer.

ii) It analyses the data in each record.

iii) It runs the FR10A program and compares these results with the results of its own analysis.

iv) It displays the results of the analysis in tabular form so that the user can browse this information in order to determine which fault records, if any, reveal problems which need to be examined in more detail. The Automatic Fault Record Analysis (AFRA) software can also display any fault record from which measurements can be taken directly.

The analysis stage in the AFRA software is carried out as follows:-

i) The particular fault recorder, its set-up parameters, its associated protection parameters and its relationship with a statistical database are accessed.

ii) This information is then passed to the expert system rule based engine.

The rules used were built up by asking experienced protection engineers what criteria they used from the analogue and digital information to determine the nature of the fault and whether there were any protection response anomalies.

The following is a typical example of the reasoning process that is undergone using the rules to determine a fault classification:-

i) a digital event occurred after an analogue disturbance (i.e. current or voltage);

ii) the residual current was abnormal;

iii) the blue phase current was abnormal;

iv) the blue phase voltage was abnormal;

v) phase voltages were present at the beginning of the record and

vi) no phase voltages were present at the end of the record.

This leads to the following fault classification:

"A blue-earth fault occurred and cut-off ensued" (suspected internal fault)

The information that is presented to the user is:

i) the time, date and fault recorder identification Which protection operated,

ii) any protection timing that is outside defined limits,

iii) the fault current duration, and

iv) the circuit breaker fault current interruption duty.

2.5.2 Circuit-breaker fault duty

The safe and reliable operation of a circuit breaker is limited by the wear on its contacts caused by its main function of breaking fault current. At present, circuit breakers are taken out of service for inspection after a set number of fault breaking operations. This set number is often based on the most pessimistic estimate of fault breaking duty. i.e. maximum fault level, and all faults associated with the same phases. In order to eliminate unnecessary circuit breaker inspections the software incorporates a fault duty monitor by automatically calculating I^2t for each phase when the circuit breaker trips. The result is put into a register where it is incremented from the last inspection. I denotes the appropriate phase current flowing during the fault and t is the time between trip relay operation and current disappearing. The I^2t formula is indicative of the interruptor contact wear. Should a more accurate formula indicative of contact wear be available it would be relatively simple to change accordingly.

2.6 Auto-Evaluation Examples

To illustrate the performance of the software, the results for three disturbances on different circuits are shown. Table 2.1 shows a reduced view of the output screen produced by AFRA. The six records correspond to the fault records from both ends of the three circuits.

Table 2.1: Typical AFRA evaluation output

File Name	Date	Time	Substation	Circuit	Classification	Protective Action
F8909602	05/10/93	16:56:07	CILF	CILF-PEMB NO 2	A yellow earth fault occurred and cut-off ensued (sup. Internal fault)	1 st & 2nd
F8909603	05/10/93	16:55:29	PEMB	CILF-PEMB NO 2	A yellow earth fault occurred and cut-off ensued (sup. Internal fault)	1 st & 2nd
F9200032	24/06/94	19:47:00	RATS	ENDE-RATS NO 2	An event occurred before any disturbance and cut-off followed (MSO?)	No action
F9200034	24/06/94	19:51:08	ENDE	ENDE-RATS NO 2	Unable to classify this record	1 st only
F9300177.SP2	30/12/95	16:27:55	SWAN	PEMB-SWAN	Minor disturbance to yellow current line	No action
F900173	30/12/95	15:33:55	PEMB	PEMB-SWAN	Minor disturbance to yellow current line	No action

The first disturbance is indicated by AFRA to be a circuit trip from a suspected internal yellow-earth fault on the Cilfynydd-Pembroke No.2 (CILF-PEMB NO 2) circuit with both first and second main protections operating. Table 2.2 shows an extract from the timing report produced by AFRA, it achieves this via a separate analysis of the digital fault record, this report shows that the protection operating times were satisfactory.

The second disturbance in Table 2.1 is indicated by AFRA to be a trip on the Enderby-Ratcliffe No.2 (ENDE-RATS NO 2) circuit, with the first main protection at Enderby being the only protection to operate, following an unclassifiable disturbance.

Detailed investigation is, therefore, needed. Further studies using FRIOA indicated that the phase comparison protection had incorrectly tripped during a load condition. (In fact the protection had mal-operated when there was no power system fault.)

The third disturbance in Table 2.1 is classified by AFRA as a yellow phase current disturbance with neither of the protections operating (In fact there was a remote external yellow fault with less than 1 kA of fault current so the protection was not required to operate.)

Table 2.2: Extract from AFRA performance report

Event number 1

Generic name: SW-OPERATED

Possible datum points from fault-logger record: The disturbance start time.

The datum time for this event is therefore calculated as 89 ms (start of fault on record)

Therefore the operation time is 99 (trip output on record) - 89 = 10 ms

The slow tolerance for this event is 17 ms

Therefore this event is within tolerance

Event number 4

Generic name: W-OPERATED

Possible datum points from fault-logger record: The disturbance start time.

The datum time for this event is therefore calculated as 89 ms (start of fault on record)

Therefore the operation time is 104 (trip output on record) - 89 = 15 ms

The slow tolerance for this event is 35 ms

Therefore this event is within tolerance

2.7 Conclusion

The use of software tools is now an integral part of the application of protection in complex situations and in the investigation of unwanted operation or failure to operate of protection systems. It provides a more accurate and efficient way of carrying out these investigations. The implementation of the Automatic Fault Record Analysis (AFRA) software will reduce the amount of effort required by a protection engineer when analysing a large number of fault records.

This example illustrates that the software has been successful in automatically evaluating protection response for different disturbance conditions correctly classifying these disturbances. Using complementary approaches, the combined software was to provide a comprehensive automatic analysis package.

However there are a number of problems with these systems in that they require complete data and can only classify faults that have been entered into the rule base. The use of Expert Systems is explored in chapter 3 and due to the nature of these systems, if employed to this problem, could handle incomplete data and attempt to classify unknown records. This best estimate could be an invaluable aid to a Protection Engineer.

Chapter III

THE USE OF EXPERT SYSTEMS IN POWER SYSTEMS MANAGEMENT

3.1 Introduction

Power systems operate most of the time under quasi-steady state conditions. Disturbances often occur in power systems, for example sudden changes in load demand, generator failure or changes to the transmission system configuration due to faults and line switching. It is important to plan and operate power systems such that [5];

- there is enough redundancy in generation and transmission capacities to cope with the changes due to these disturbances
- there are enough controllable resources (generation re-dispatch, load control, etc).

Such 'robustness' of the system relative to imminent disturbances is referred to as the security of the system. The concept of security in power system operation may be configured into three components, namely monitoring, assessment and control, under the following frame work of operation, we have to operate within the bounds of economics and constraints;

Security monitoring

(i) Using real-time system measurements (state estimation) identify whether the systemis in a normal state or not. If the system is in an emergency state, go to step (iv). If load has been lost, go to step (v).

(ii) Security assessment

If the system is in a normal state, determine whether the system is secure or insecure with respect to a set of next credible contingencies.

(iii) Preventive control

If the system is insecure, i.e. one or more of a set of scenarios that can cause an emergency situation have been identified, determine what preventive action should be taken to make the system secure.

(iv) Emergency control

Execute the proper corrective action to return the system to a normal state.

(v) Restorative control

Restore service to system loads.

Security analysis and control have been implemented in modern Energy Control Centers (ECC), by a software package called *Energy Management Systems* (EMS). EMS contains powerful Power Application Software (PAS) like state estimation, scenario analysis, operator load flow, short circuit analysis, stability analysis, optimal power flow with security constraints, etc. Figure 3.1 shows the major components of an EMS.

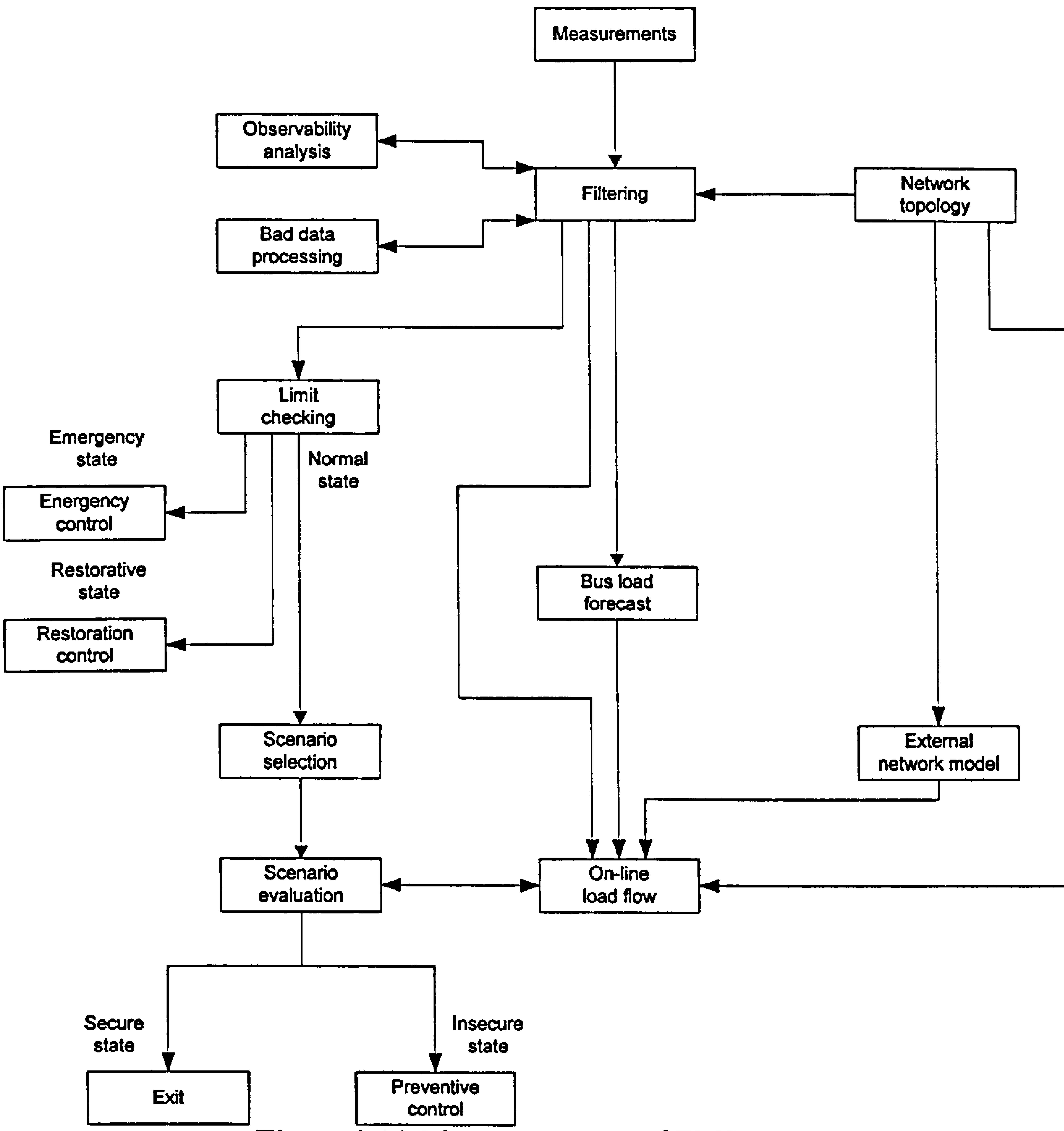

Figure 3.1 Major components of EMS

An estimate of the system state variables (bus voltage magnitudes and phase angles, for normal steady state) is obtained by *state estimation.* The line loading and other information concerning system operation may be derived from the estimated state variables. State estimation provides the real time database for other analysis, control and optimization functions. To assess whether a normal operating state is secure or not, the system has to be subjected to contingency analysis. Contingency analysis is a software application run in EMS to give the operators an indication of what might happen to the power system in the event of a probable equipment outage.

3.2 Power System Security

Power system security can be classified into six different levels [5];

level 1: Secure.

level 2: Correctively Secure.

level 3: Alert.

level 4: Correctable Emergency.

level 5: Non-correctable Emergency.

level 6: Restorative.

The block, schematic of power system static security levels is shown in Figure 3.2. The arrowhead lines represent Involuntary transitions between levels 1 and 5 due to scenarios. The static security level of a power system is characterized by the presence or otherwise of emergency operating conditions (limit violations) in its actual (pre-scenario) or potential (post-scenario) operating states. Levels 1 and 2 in Figure 3.2 represent normal power system operation. Level 1 has ideal security: the power system survives all the credible scenarios without relying on any post-scenario corrective action. Level 2 is more economical, but it depends on EMS corrective actions to remove violations without loss of load, within a specified period of time. The removal of violations from level 4 generally requires EMS directed "corrective rescheduling" bringing the system to level 3. Once level 3 has been reached, further EMS directed "preventive rescheduling" must be performed to return the system to either level 1 or 2. If the power system has reached level 5, load will be lost by automatic local switching actions or-by commands from the control center. To obtain corrective and preventive control strategies, fast computations become necessary in ECC. Here scope exists for expert system application as they provide fast solutions with minimum of numerical computations.

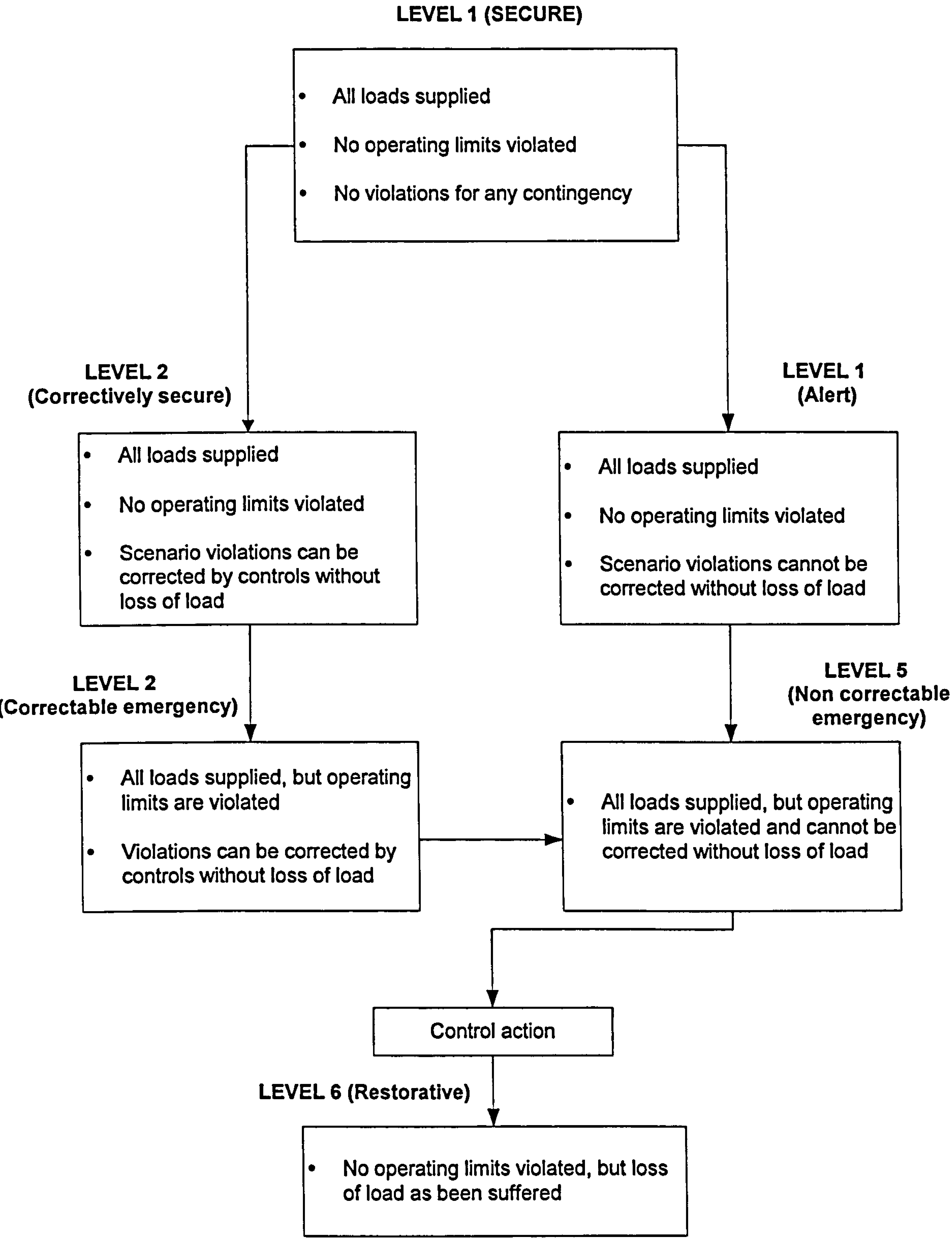

Figure 3.2 Power system static security levels

3.2 The Need for Expert Systems application to Power Systems

The objective of an Energy Control Center (ECC) is, to ensure secure and economic operation of power system. The challenge to optimize power system operation while maintaining system security and quality of supply to customers is increasing. Globally, growing demand without matching expansion of generation and transmission facilities and more tightly interconnected power systems contribute to the increased complexity of system operation. Rising costs and environmental concerns have made transmission, as well as generation, systems to be operated closer to design limits with smaller security margins and hence greater exposure to unsatisfactory operating conditions following a disturbance. As system complexity increases, the ability of an operator in ECC to respond correctly to unforeseen events in the system correspondingly reduces. This growing complexity is causing problems like;

- A rapid increase in the number of real time messages, has made operator response more difficult, due to the human cognitive barrier.

- Current numerical processing software being computationally intensive is found inadequate to meet the operational requirements of power systems, especially during emergency conditions.

- Most design, planning and control problems encountered are complex and time consuming, because of multiple objective functions, multiple constraints, complex system interaction and so on.

- The operator has to continuously handle three questions, given the present state of the power system:

  i. What might be done to improve system efficiency, reliability and to reduce stress on equipment?

  ii. Can the system withstand a first or perhaps a second scenario?

  iii. What will be the state of the power system in the near future after planned outages and generation changes occur?

Most of the network security analysis and optimization functions are computationally intensive. Table 3.1 shows typical CPU time and I/O time requirements for various network analysis functions and the frequency of execution in practice for a 500 bus system on a one MIP (million instructions. per second) machine [5, 6]. These functions are executed sequentially at specified time intervals. Generally state estimation is carried out once every 5 minutes, DC contingency analysis which is fast, but approximate once every 10 minutes, AC contingency analysis once every 20 minutes and optimal power flow once every 30 minutes. Typical scenarios on a power system consist of outages such as loss of generating units, transmission lines, transformers, etc. scenarios are selected based on the criteria;

- A scenario is likely to occur
- A performance index (PI) of plant and transmission lines
- Ranked a scenarios based on the values of PI

Depending upon the accuracy needed and CPU time available DC or AC scenario analysis is performed. If a power system has serious reactive power flow or voltage problems an AC power flow solution is necessary for accurate results. These contingencies may cause network overloads, voltage limit violations or stability problems.

Table 3.1: Network analysis computer load (1 MIP CPU, 500 busses)

PAS function	CPU time (s)	I/O time (s)	Frequency (min)
State estimation	30	10-20	5-10
AC security analysis	20-25	3-5	10-20
DC security analysis	3-15	3-5	5-15
Voltage scheduling	70-80	5-10	15-30
Economic dispatch	10-15	3-5	5-10
Security dispatch	22-30	5-8	15-60
Operator load flow	10-20	5-10	On request
Optimal load flow	50-80	5-10	30-60

It has been found that many important PAS tools are not fully used because under operating conditions they are of restricted importance, while before or during disturbances it is difficult to include them in the operators decision-making process. Current control system software in ECC have been developed primarily to facilitate monitoring and control of power systems under normal/alert states. Functions such as state estimation, security analysis and optimal power flow are used to ensure secure operation of a power system.

Functions such as automatic generation control, economic dispatch, unit commitment and load forecasting are used to ensure optimum economic operation of a power system. Currently available EMS do not provide the operator with support as to how and when to use which PAS software functions. Nor do they offer any interpretation of the results with respect to the actual operating state. The EMS uses complex algorithms, which due to the high dimensionality and the non-linear characteristics of a power system, cannot exhaustively search for the best or for the correct response. These functions can presently only provide partial solutions which have to be used by the operator to determine the best overall solution. To take correct action implies a very complex decision making process, often resulting in an unacceptable time pressure on the human operator.

In order to reduce the possibility, extent and duration of major failures in power systems preventive, corrective and restorative actions are required. The control system software in ECC should be expanded to include developing preventive and corrective control strategies for different scenarios. Scenarios may be of single type (single equipment outage) or double type (overlapping of two single scenarios). For an interconnected power system this may lead to a very large number of scenarios. Processing all these credible scenarios is emerging as the most serious computational bottleneck for on-line control in an ECC. With these constraints present day PAS functions in EMS may not be suitable for on-line application for power system control.

This limitation of modern EMS can be overcome with the integration of expert systems into an EMS. Expert systems give very fast, though approximate, but acceptable solutions as they mostly use symbolic processing. The expert system can potentially provide a rapid reaction to emergency events by summarizing information quickly and checking many more applicable rules than a human operator could in the same period of time. The expert system can be used in any one of the following modes:

Training simulator: To train utility personnel in power system operation and control in off-line mode.

Preventive mode: Generate preventive control strategies for credible contingencies. The use of expert systems ensures faster processing of contingencies

Corrective mode: Generate real time control actions for a scenario which has occurred.

The output of the expert system is presented in the form of data and advice to the operator, who accepts, modifies, or ignores them using his engineering experience (based on utility practice).

3.3 Expert Systems

An expert system [7, 8, 9, 10] is a set of computer programs that use current information, a knowledge data base, and inference rules to solve problems that are difficult enough to require significant human expertise for their solution. To do so, it simulates the human reasoning process by applying specific knowledge and inferences. Figure 3.3 shows a general architecture of an expert system.

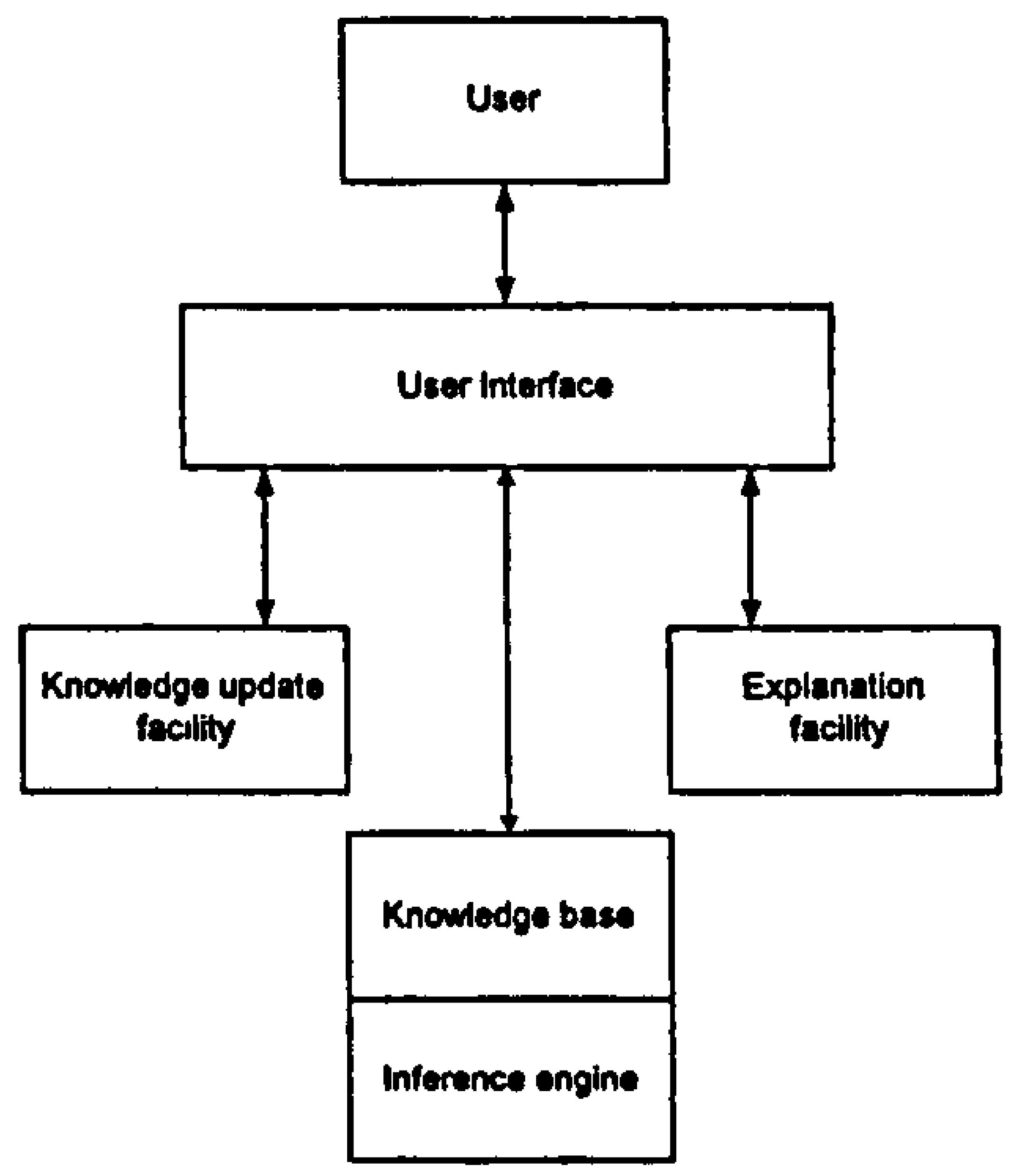

Figure 3.3 General architecture of expert systems

An expert system seeks a solution, one that is good enough even though it may not be an exact optimum solution. Expert systems methodology seems applicable to solve problems for which,

- No analytical method is known.
- The analytical method is partially known.
- The analytical method is known but very inefficient in terms of providing a solution for on-line application.

If the expert system has to he applied, 3 conditions must be fulfilled:

1. Human expertise must be available for providing domain specific knowledge,

2. Expert system based search and inference have to be fast and reliable, and

3. The expert system must be capable of explanation and justification.

3.4 Integration of Expert Systems with EMS

Algorithmic programs efficiently solve problems for which rigorous analytical solutions exists, while expert systems are much better at handling heuristic knowledge. Since solving practical problems often requires both types of abilities, integrating expert system modules with numerical modules in EMS results in a hybrid system with enhanced features. The different types of integration methods are:

- using communication files.
- using function calls.
- using distributed configuration.
- integration with a data base management system.

3.4.1 Communication Files

In this mode of integration, all the communications between numerical and expert system modules are performed via files as shown in figure 3.4. This scheme is relatively simple to implement but provides only a loose integration.

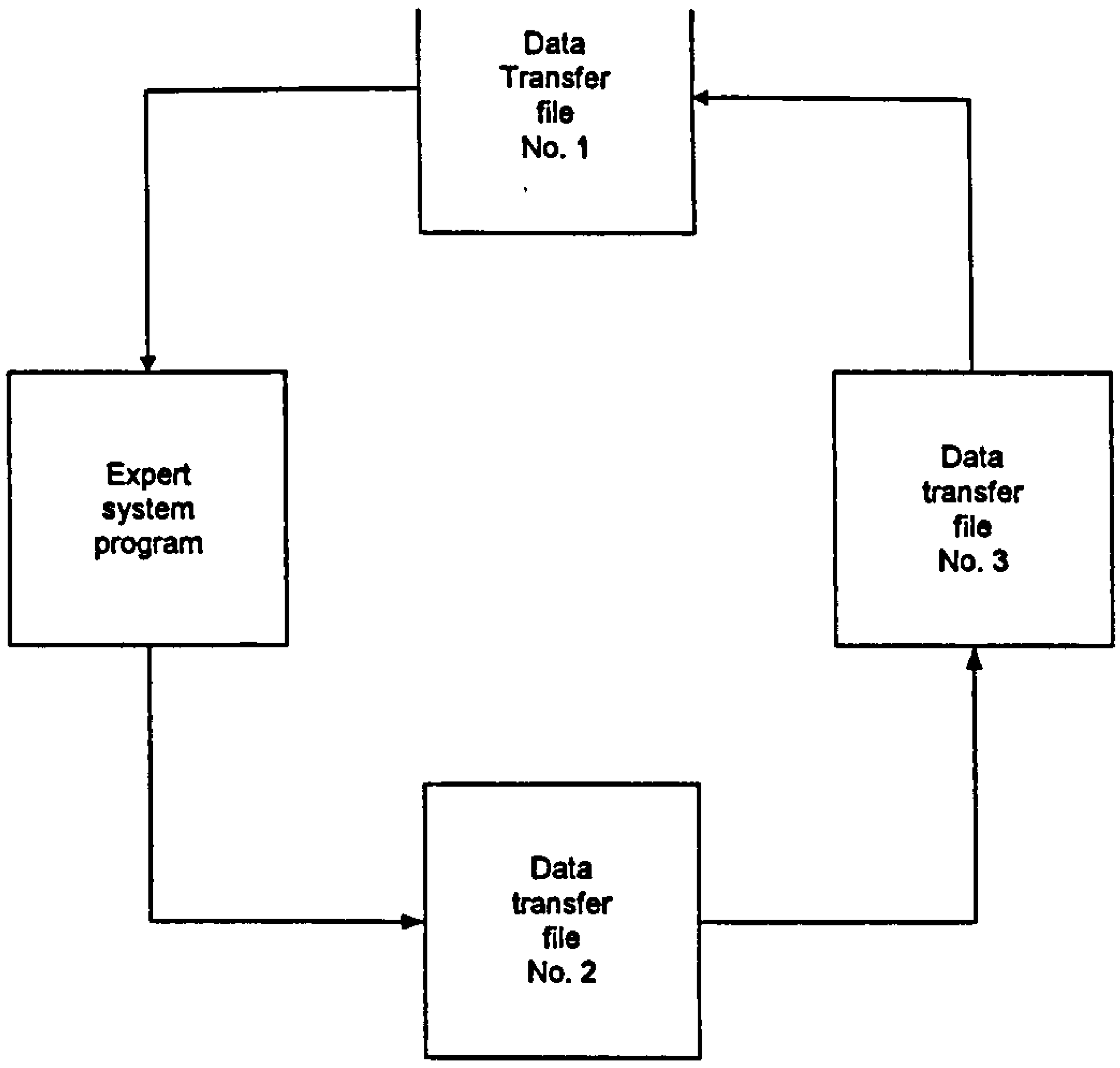

Figure 3.4 Communication files for integration

3.4.2 Function Calls

Some Expert systems are being developed using conventional languages like C and Fortran. It is thus possible to link expert system modules with numerical modules in a single executable file as shown in figure 3.5. This approach allows a much tighter integration.

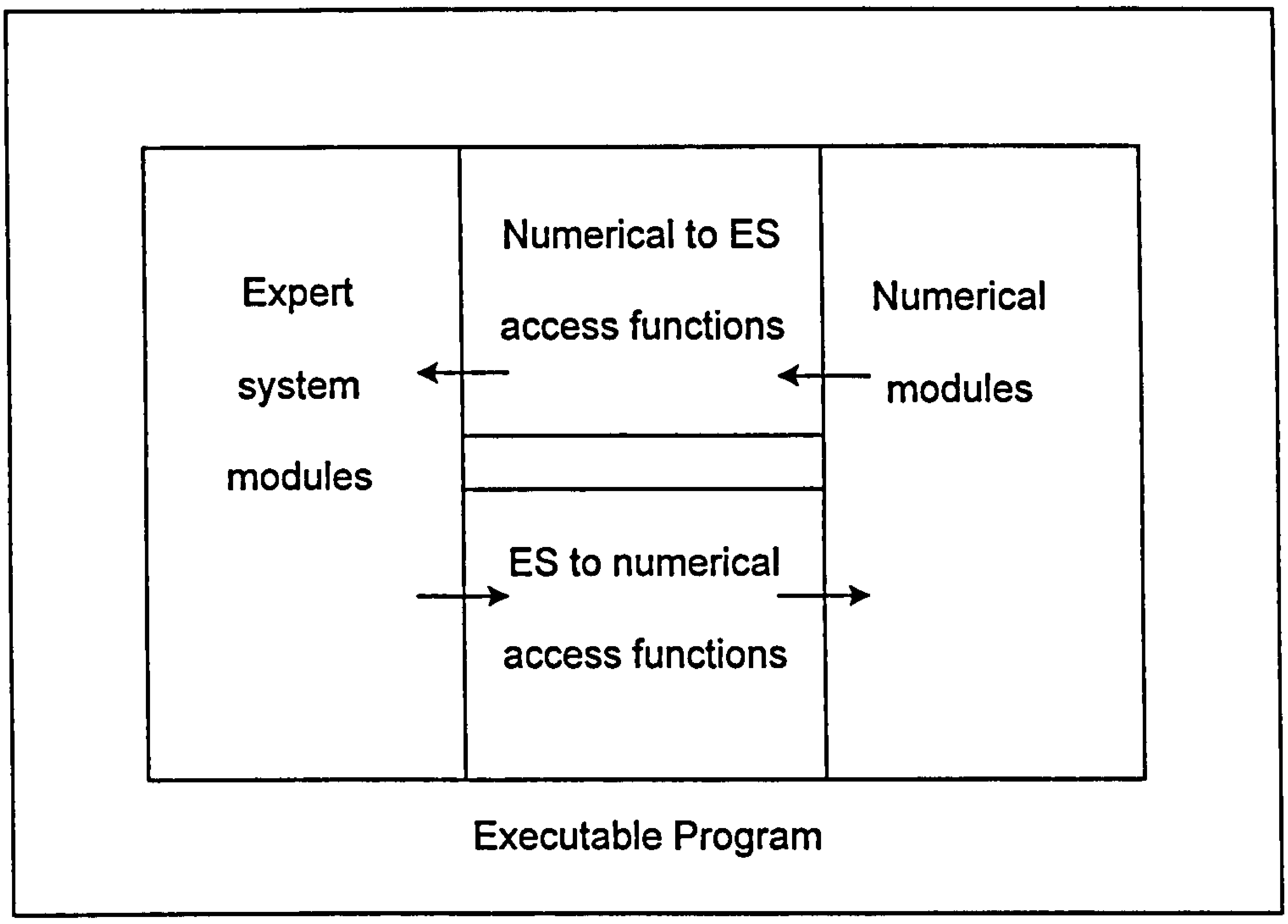

Figure 3.5 Function calls for integration

3.4.3 Distributed Configuration

The expert system modules and numerical modules can also run as separate asynchronous processes. Communication between them relies on the inter-process communication mechanisms provided by the operating system, such as message queues under UNIX. Furthermore, these processes are not limited to running on the same computer. they can be run on different machines and communicate via a Local Area Network.

3.4.4 Integration with a Data Base Management System

Database management systems are a special type of conventional programs. If tools are available for mapping a database into the database structure used by expert systems, then this type of integration can he used.

3.5 Advantages of Expert Systems

From the literature survey [11, 12, 13] the following have been identified as the main advantages of an expert system:

- An expert system can perform at a level exhibited by a recognized expert in the problem domain. This leads to automation of routine operational tasks and provides the system operator in ECC an environment that enhances his productivity, thus leading to efficient operation.

- Expert systems provide a better and faster solution to emergency events than human, operators. This is very useful in power system operations.

- Each production rule represents a piece of knowledge relevant to the task. Hence it is very convenient to add, remove and modify a rule in the knowledge base as experience is gained.

- Expertise becomes more widely available for the utility and can be used for a variety of purposes. For example, the expert system can be employed as a tool for training new personnel.

- A major concern for electric utilities is the loss of expertise due to retirement of experienced operators. To help maintain the experience gained over a number of years, knowledge bases can be used as depositories of human knowledge. An expert system can use these knowledge bases to assist the operators during day-to-day and emergency operation of the power system.

- Human experts, especially the highly skilled ones, are very scarce, and hence very expensive. Expert systems in contrast are relatively inexpensive (They are costly to develop but inexpensive to operate).

- A successful expert system implementation provides the utility with a system that can provide consistent reliable performance that will not degrade in stressful situations and is available round-the-clock.

There are important areas in which human expertise is clearly superior to expert systems. This does not reflect a fundamental limitation of expert systems or artificial intelligence, but just the current state-of-the-art. Two major limitations of expert systems are:

- Humans are much more creative and innovative than even the smartest programs.

- Human experts excel in learning. Human experts adapt to changing conditions, they adjust their strategies to conform to new situations. Expert systems are not particularly adept at learning new concepts or rules, which is a stumbling block for artificial intelligence.

An ideal expert system can be characterized by:

- Extensive specific knowledge from the domain of interest.
- Application of search techniques.
- Support for heuristic analysis.
- Capacity to infer new knowledge from existing knowledge.
- Symbolic processing.
- An ability to explain its own reasoning.

3.6 Expert System application to Power Systems

Figure 3.6 shows a block schematic of integration of expert systems into an EMS.

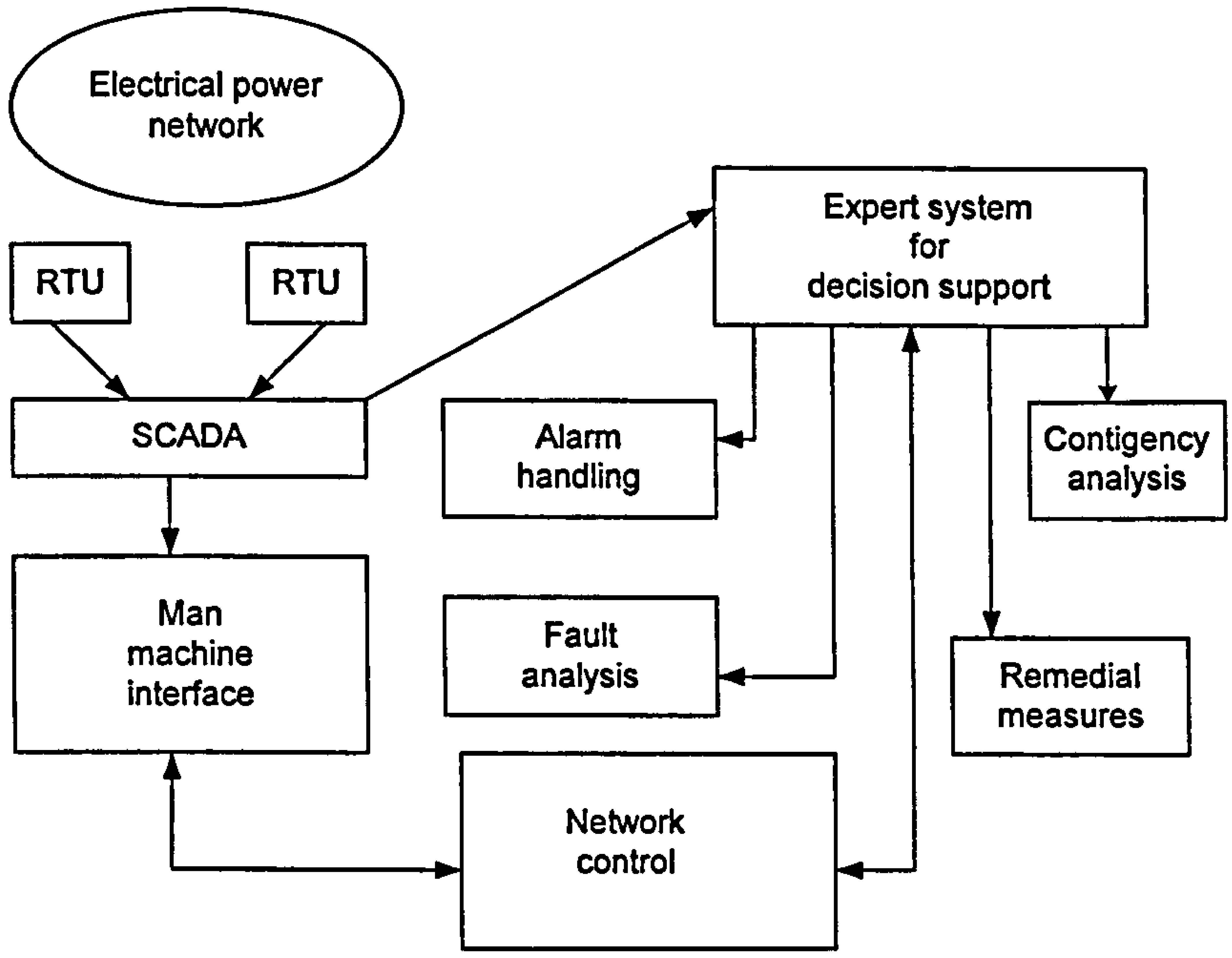

Figure 3.6 Expert system as a part of EMS

The functions of expert system shown in figure 3.6 include:

- Continuous evaluation of the system state with respect to the classifications 'normal', 'alert' and 'disturbed'.
- Suggestion of feasible preventive or corrective action with respect to the present system operating state.
- Information compression in case of a network disturbance including fault analysis and fault location.
- Proposals for adequate load management.
- Evaluation-of switching operations proposed by the operator.

From the literature survey [14], the most commonly investigated expert system areas are:

- Alarm reduction.
- Fault diagnosis.
- Steady state and dynamic security assessment.
- Restoration.
- Remedial actions.

- Substation monitoring and control.
- Maintenance scheduling.
- Expert system development tools.

In some cases, the advent of expert systems has resulted in the creation of new power system tools, where rigorous tools were previously unavailable (for example alarm processing, fault location and automated restoration). In other cases expert systems relying on past experience and logical reasoning as a basis for decision making, are competing with numerical tools (for example unit commitment, network switching and load forecasting). In many recent applications, expert systems are providing both the logical control for calls to real time data acquisition and numerical algorithms (load flow, stability analysis, reliability, etc) and the expertise to interpret and to further process information.

3.6.1 Alarm reduction

In an emergency condition, existing EMS tend to generate numerous alarms as a result of discrete and analog data received by SCADA (Supervisory Control And Data Acquisition). Some alarms may be false, others may be missing. In such an environment, the system operator needs to understand the alarms, determine what events lead to these alarms and take appropriate actions to restore service to customers. Expert systems are used to filter these alarms and interpret them, thereby providing the system operator with a more accurate picture of the likely events that might have caused the alarm. Several different approaches to developing expert systems have been used like:

- purely rule based systems.
- frame based systems.
- hybrid systems which are a combination of rule based and frame based systems.

[15] summarizes the results of an international survey on different utilities approach to alarm reduction. A major difficulty that remains is development, of a suitable interface between expert systems and the real time database of the ECC.

3.6.2 Fault diagnosis

The purpose of fault diagnosis is to identify the sequence of events which lead to protection relay and circuit breaker operations. E. Cardozo and S.N. Talukdar [15] have developed a rule-based system capable of identifying the fault location and malfunctioning relays/breakers in a power system. K. Komai et al [17] combine heuristic rules which help reduce the possible scenarios, thereby reducing the number of fault cases that need to. be simulated. C. Fukui and J. Kawakami [18] consider multiple faults in their expert systems, which has been implemented in Prolog. K. Tomsovic et al [19] propose an expert system in OPS83 called CRAFT (Customer Restoration And Fault Testing) which is able to identify the faulted section of a multi-tapped transmission line equipped with automatic switches and breakers. Y. Sekine et al [20] have presented a comprehensive view about the accomplishments, limitations and effectiveness of expert system implementation for fault diagnosis of power systems. One major limitation of expert systems developed is their inability to deal with complex scenarios with multiple faults and equipment failures.

3.6.3 Restoration

The restoration of a power system involves re-energizing customers interrupted during a disturbance. Various constraints need to be considered like:

- availability of generation and transmission facilities.
- availability of data acquired from communication systems.
- line flows and bus voltage limits.
- system stability.

T. Sakagtichi and K. Matsumoto [21] presented the first paper on expert system application to power system restoration. The knowledge base of the expert system contains rules which can identify paths of energisation for de-energised busses. Y. Kojima, et al [22] developed an expert system that selects the target system during restoration and provides guidance in switching actions and load dispatching. Y. Kojima. et al [23] also reported development of an expert system which performs a two phase task, energizing (restoration of substations and plants) and load supplying (re-energizing loads without overloading equipment). Their expert system incorporated detailed substation models and utilized system change-over bus splitting and load transfer to relieve overloads. K. Tomsovic et al [19] have developed an expert system CRAFT which suggests switching and breaker action aimed at restoring the largest number of substations served from the line. Almost all approaches are based on searching for energisation paths and feasibility check with a power flow program. In practice, a black out could lead to confusion in ECC due to wrong or missing information. A more robust expert system capable of dealing with uncertainties and erroneous data is desirable.

Bell et al [24] report on a Model-based diagnostic system that provides automatic analysis of the available fault recorder data and SCADA data, and provides an intelligent decision support system.

3.7 Conclusion

The development of expert systems is a continuous process as new knowledge is gained in the field of artificial intelligence and new expert system development tools are built. Efforts are being made for on-line application of expert systems in ECC as preventive control under normal/alert conditions and as a corrective control during a disturbance. This will enable a more secure power system operation. Considerable scope exists in the development of expert systems and their application to power system operation and control.

Chapter IV

NEURAL NETWORKS

4.1 Introduction

In the late 1980's an area called Connectionism, Neural Networks or Parallel Distribution Processing emerged as a topic for research and for commercial development. Neural Networks were not entirely a new field [25, 26, 27], what was new was the concern with well-founded analysis and a deepening understanding of the topic. The topics mentioned all refer to machines that unlike conventional computers have a structure that reflects the structure of the brain. Biological computing, the brain and the nervous system of animals and human beings have existed for millions of years. They are effective in processing sensory information and controlling the interactions of animals within their environment.

The brain is able to cope with sophisticated recognition and inductive tasks far beyond the capabilities of systems based on present computing logic and architecture. A simple example is when a person sees an old friend; he or she does not search a database of stored images of all the people he or she knows, the recognition if it happens is almost Instantaneous. Another example is the recognition of speech; a human brain can recognize speech even if there are distortions such as noise and accent, where as modeling speech becomes very difficult.

The key point is that there are no obvious rule sets that can equal the performance of the brain in many important tasks. Humans find these tasks quite easy and all they have to go on is their experience stored in a network of interconnected brain cells called neurons. The fact that biological computation is so effective suggests that it may be possible to attain similar capabilities in artificial devices based on the design principles of neural systems.

Physicists, biologists, mathematicians, computer scientists, statisticians and many others have contributed to the research of Neural Networks. The investigations started back in 1943 by McCulloch and Pitts [28] who investigated neurons as logical devices. In the 1960s Rosenblatt [29] created adaptive neurons and simple networks that learn. In the 1970s there were many smaller investigation carried out by investigators such Carpenter and Grossberg [30,31]. They attempted to model the behavior of real neurons in computation networks more closely and to develop mathematics and architectures for extracting features from pattern, for classifying pattern and for associative memory in which information itself serve to retrieve an entire memory.

In the 1980s neurobiologists were gaining more understanding of how information is processed in nature. Cheap computer power made it possible to analyze the models in detail and new concepts in the mathematics of Neural Networks accompanied developments.

One fascinating property of Artificial Neural Networks is their ability to exceed the limitations of traditional information proceeding such as the need for detailed programming. Neural Networks have many interesting properties such as their capacity for adaptation, use of distributed memory, capacity for generalization, ease of construction and their parallelism.

Although Artificial Neural Networks do have several fascinating properties they also have several limitations. Most networks are simulated on sequential machines which means as the problem gets larger the processing time rapidly Increases. The main disadvantage is their inability to explain any results obtained. The quality of their performance is measured by statistical methods.

Problems which are suitable for implementation by Artificial Neural Networks include pattern recognition, signal processing, Vision, speech processing, forecasting and modeling decision making aids and robotics. Artificial Neural Networks are used in industrial applications, the financial sector, the telecommunications sector and the environment sector. Table 4.1 gives examples of the applications in each sector.

Table 4.1: Applications of Neural Networks by Sector

Sector	Applications
Industry	Quality control, production planning and fault diagnosis.
Financial	Prediction and modeling of market and signature analysis.
Telecommunications Signal	analysis, noise elimination and data compression
Environmental	Risk evaluation, chemical analysis and weather forecasting

4.2 The Operation of a Neural Network

A Neural Network is modeled on the gross structure of the brain: a collection of nerve cells or neurons, each of which is connected to as many as 10,000 others, from which it receives stimuli - inputs and feedback - and to which it sends stimuli. Some of those connections are strong; others are weak. The brain accepts input and generates responses to them, partly in accordance with its genetically programmed structure but mainly through learning, organising itself in reaction to input rather than by doing only by rote what it is told to do.

The Neural Networks used by engineers are only loosely based upon biology, they only behave in a vaguely similar way. Since there is little understanding of how the brain works, it will be a long time before a machine can re-create all the capabilities of the human brain. Even so, Neuro-computing is already offering some valuable, specialised, brain like capabilities that in all likelihood lie beyond the reach of algorithmic programming. Neural Networks are varied. Many different types are being explored in research and many common ones that are being used. Table 4.2 shows various classifications of neural networks according to their architecture and processing mode. Fig 4.1 illustrates the types of neural networks used as classifiers.

Table 4.2: Classifications of Neural Networks According to their Architecture and Processing Mode

Network Architecture	**Processing** Feed Forward (FF) Recurrent (REC)	**Method** Continuos Time(CT) Discrete Time(DT)
Single-layer Network of Discrete and Continuous Perceptrons	FF	-
Multilayer Network of Discrete and Continuous Perceptrons	FF	-
Linear Associative Memory	FF	-
Associative Memory (Hopfield Net)	REC	DT or CT
Hamming Network	FF	-
MAXNET	REC	DT or CT
Clustering Network	FF	-
Self-Organizing Neural Array (Kohonen Net)	FF	-
Adaptice Resonance Theory 1 Network	REC	DT or CT

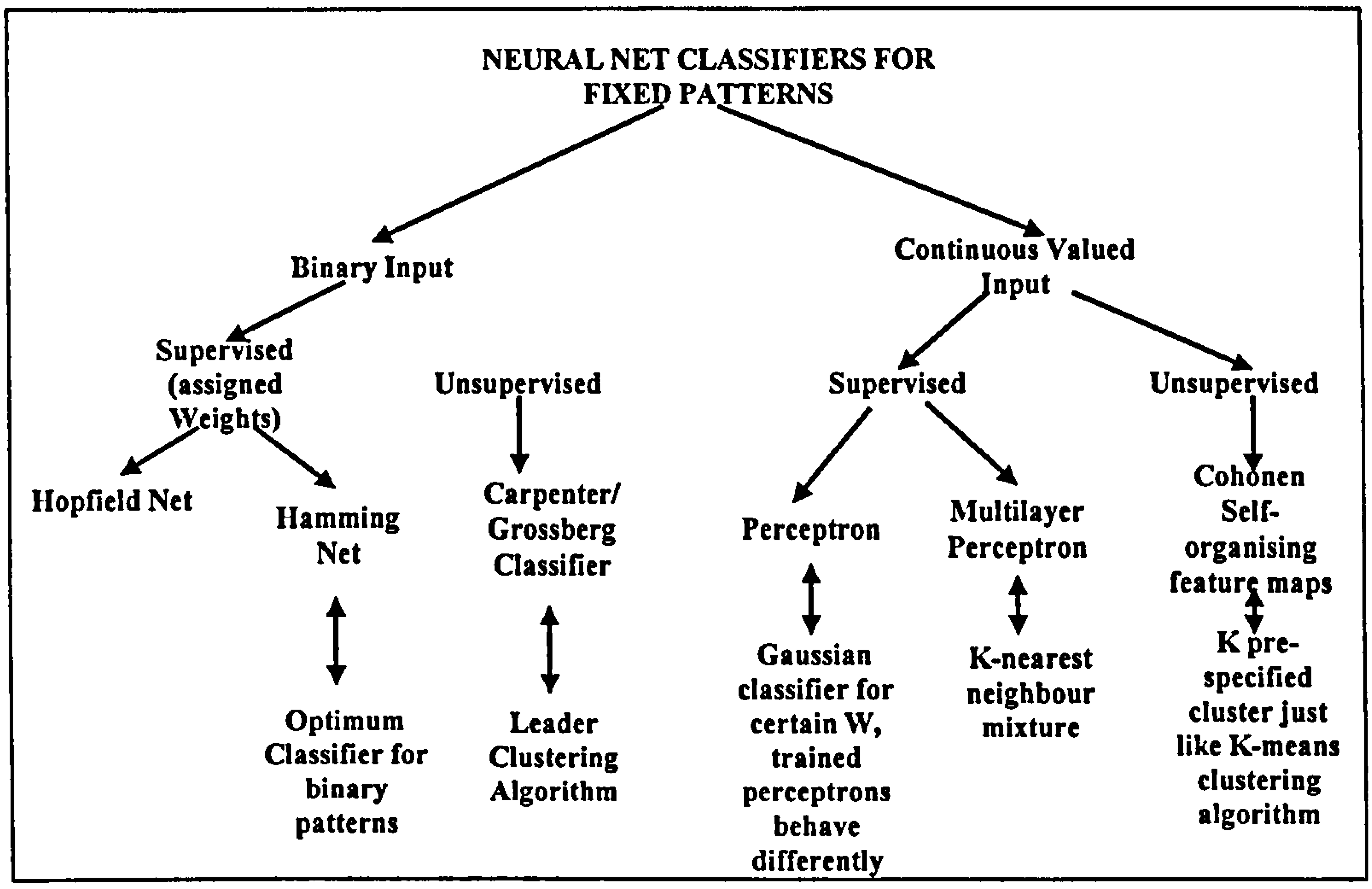

Figure 4.1: Types of neural networks used as classifiers

4.3 Uses of the Neural Network

This section describes a number of current uses of neural networks

- Classification: an input pattern is passed to the network and the network produces a representation class as output.
- Pattern Matching: an input pattern is passed to the network and the network produces the corresponding output pattern.
- Pattern Completion: an incomplete pattern is passed to the network and the network produces an output pattern, that has the missing portion of the input pattern filled in.
- Noise-Removal: a noise-corrupted input pattern is presented to the network and the network removes some (or all) of the noise and produces a cleaner version of the input pattern a output.

- Optimisation: an input pattern representing the initial value for a specific optimisation problem is presented to the Neural Network and the network produces a set of variables that represents a solution to the problem.
- Control: The input to the neural network will be the current state of a system and its desired state. The output of the neural network will give a command to the controller to achieve the desired state.

4.4 The Neuron

In order to understand the computational abilities of the brain it is necessary to understand the basic structure and functions of the neuron. Nerve cells, called neurons, are the fundamental elements of the central nervous system. Neurons in the brain are arranged in quite regular pattern and are grouped in functional divisions. The central nervous system is made up of one hundred billion neurons, these neurons have a number of distinct characteristics but there are no two neurons exactly alike. The structure of the brain is also highly varied from one individual to another.

An artist's impression of a neuron is shown in Fig 4.2.

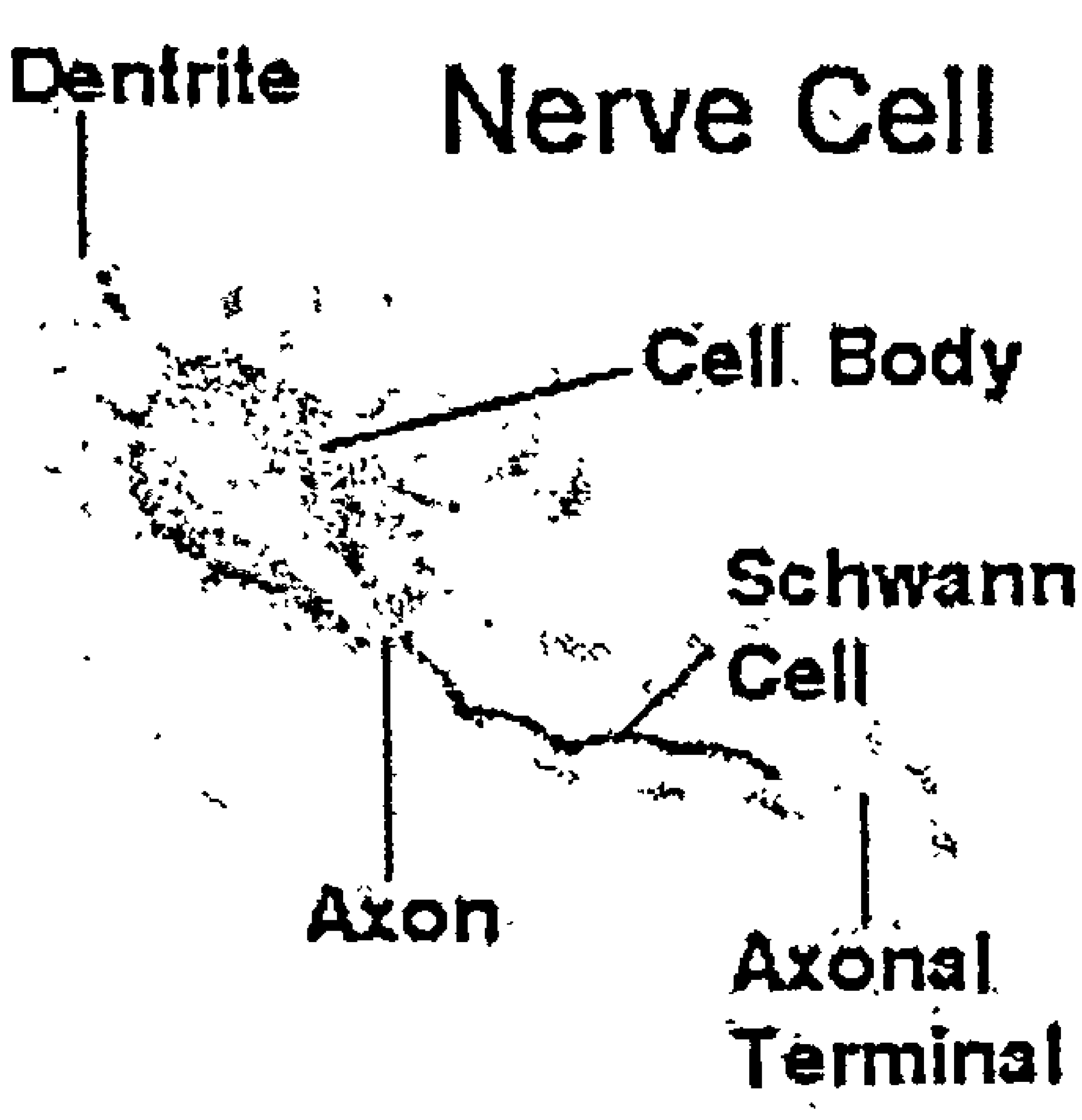

Figure 4.2: An artist's impression of a neuron

The cell body contains the nucleus of the neuron, its shape is usually a pyramid or a sphere but this depends on the position of the brain. The cell body is some microns in diameter.

Every neuron has around it a structure of dendrites, which are fine tubular extensions, these dendrites are some tenths of a micron across and tens of a micron in length. The dendrites are the principal receptors of the neuron and serve to connect its incoming signal. The axon, also referred to as nerve fiber, is the outgoing connection for signals emitted by the neuron. The axon varies from a millimeter in length to more than a meter in length.

Neurons are as intricately connected as a bramble patch overgrown with vines, they are very tiny and delicate and it is hard to measure what is happening to them without damaging them or interfering with the flow of information related to their operation. Neurons are connected to form the central nervous system, the connection among two neurons takes place at the synapses where they are separated by a synthetic gap of the order of one hundredth of a micron. The connection between the neurons mediates the strength with which a signal crosses from one neuron to another. Simple electronics components can be used to build a network, operation amplifiers can be used to replace the neurons and wires, resistors and capacitors replace the connections.

The functions performed by a neuron depends on the properties of its external membrane; this fulfills five functions

(i) It serves to propagate electrical pulses along the length of the axon and of its dendrites.

(ii) It releases transmitter substances at the extremity of the axon.

(iii) It reacts with the transmitter substances in the dendrites at the cell body.

(iv) It reacts to the electrical impulses that are transmitted from the dendrites and generates or fails to generate a new electrical pulse.

(v) It enables the neurons to recognize which other neuron it should be connected to.

To summarize, neurons receive signal coming from neighboring neurons, integrate these signals, give rise to these pulses, conduct these pulses and transmit them to other neurons that are capable of receiving them.

4.5 The Structure of Connections

Biological studies of the brain have shown that the number of connections between neurons is enormous. The cortex is divided into a number of different layers. In one layer the number of interactions is large but neurons in one layer are also connected to other layers. Generally a network may be totally connected where all neurons are connected to each other or locally connected where neurons are connected only to their nearest neighbors.

4.5.1 A Fully Connected Network

Each layer receives signals from the previous layer and transmits results to the next layer. A fully connected network is one where every neuron is connected to every neuron including itself. There are feed-forward and feed-back networks. A Feed-forward network (associative) is one in which there are no closed loops and which acts between a set of overall input terminals and a set of overall output terminals by learning to associate patterns at the former with the latter. A feedback network is one where information can find its way around a loop from the output back into the input.

The associative mode of a network can be contrasted with the auto associative style of a network in which a network learns to associate an output with a given input and to produce roughly the right output even if the input is slightly distorted.

A processing element is the portion of the Neural Network where all the, computing is performed. A processing element function maps a processing element domain to a pro specified range. The number of processing element functions are infinite but five are regularly employed by the majority of Neural Networks, they are: -

(i) Linear Processing element function

(ii) Stop Processing element function

(iii) Ramp Processing element function

(iv) Sigmold Processing element function

(v) Gaussian Processing element function

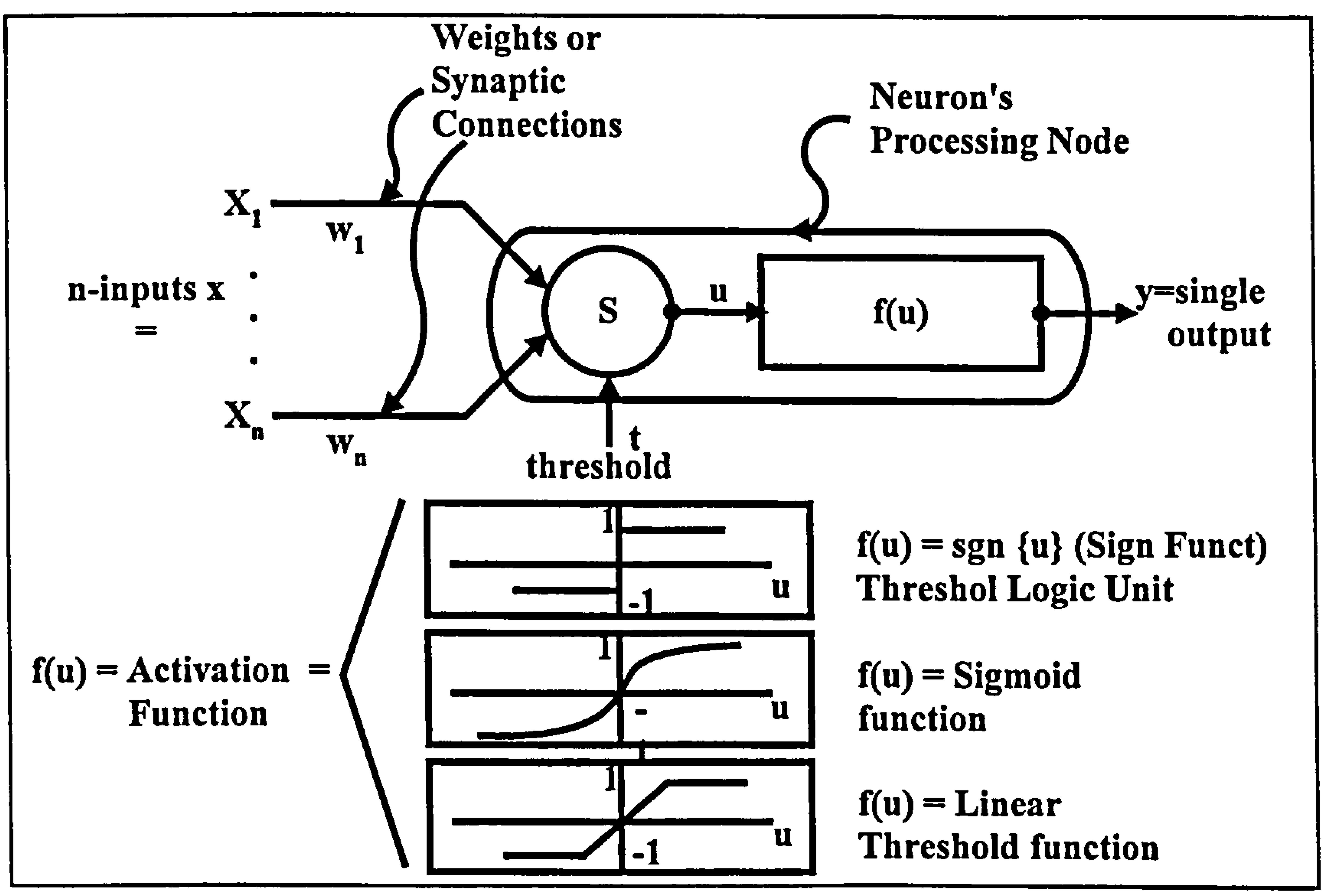

Figure 4.3: Mathematical model for the working principle of a neuron

Figure 4.3 is a mathematical model for the working principle of a neuron

A Neural Network is equivalent to a directed graph. A directed graph has edges (connected) between nodes (processing elements) that allow information to flow in only one direction shown by the arrow. Neural Networks extend the directed graph to include a weight at each edge or connection that modulates the amount of output signal passed from one processing element down to the adjacent processing element. These connection weights are adjusted during the learning process.

4.6 Memory and Learning

Recent research shows that the growth of a particular nervous system follows a hereditary program. When a child is born the nervous system is defined to be a network of neurons with a full set of connections. The development of the nervous system is due to the interaction between the external environment and the genetic program.

The development of a network takes place by means of a selective mechanism that inputs an image of the environment onto the network. Learning and memory development can be characterized simply by changes in the connection between neurons.

During the learning process in a Neural Network the connection weights are adjusted. Connection weights that are positive values are excitatory connections and those with negative connections are inhibitory connections. A connection with a weight of zero could mean that no connection exists.

The most appealing quality of a Neural Network is its ability to learn. Learning is defined as a change in connection weight value that results in the capture of information that can be later recalled. There are several procedures available for changing the value of the connection weights.

All learning methods can be classified into two categories, supervised learning and unsupervised learning. Figure 4.4 illustrates the flow of information in each method.

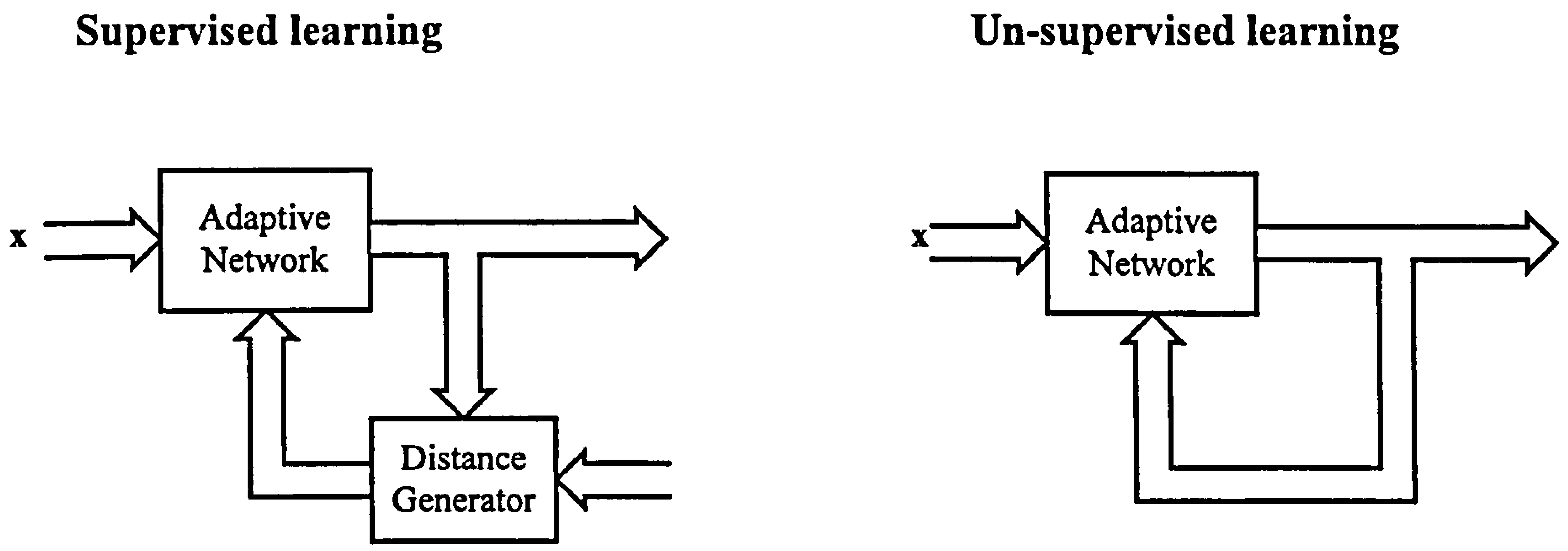

Figure 4.4 Flow of information for Supervised and un-supervised learning

4.6.1 Supervised Learning

This is a process that incorporates an external teacher and or global information Examples of supervised learning include error correction learning, reinforcement learning and stochastic learning. Supervised learning can be separated into two sub categories, structural learning and temporal learning. Examples of structural learning include pattern matching and pattern classification. Temporal learning is concerned with capturing a sequence of patterns necessary to achieve some final outcome. In temporal learning the current response of the network is dependent on previous responses, examples of structural learning include prediction and control. Figure 4.5 illustrates the flow of information during supervised learning.

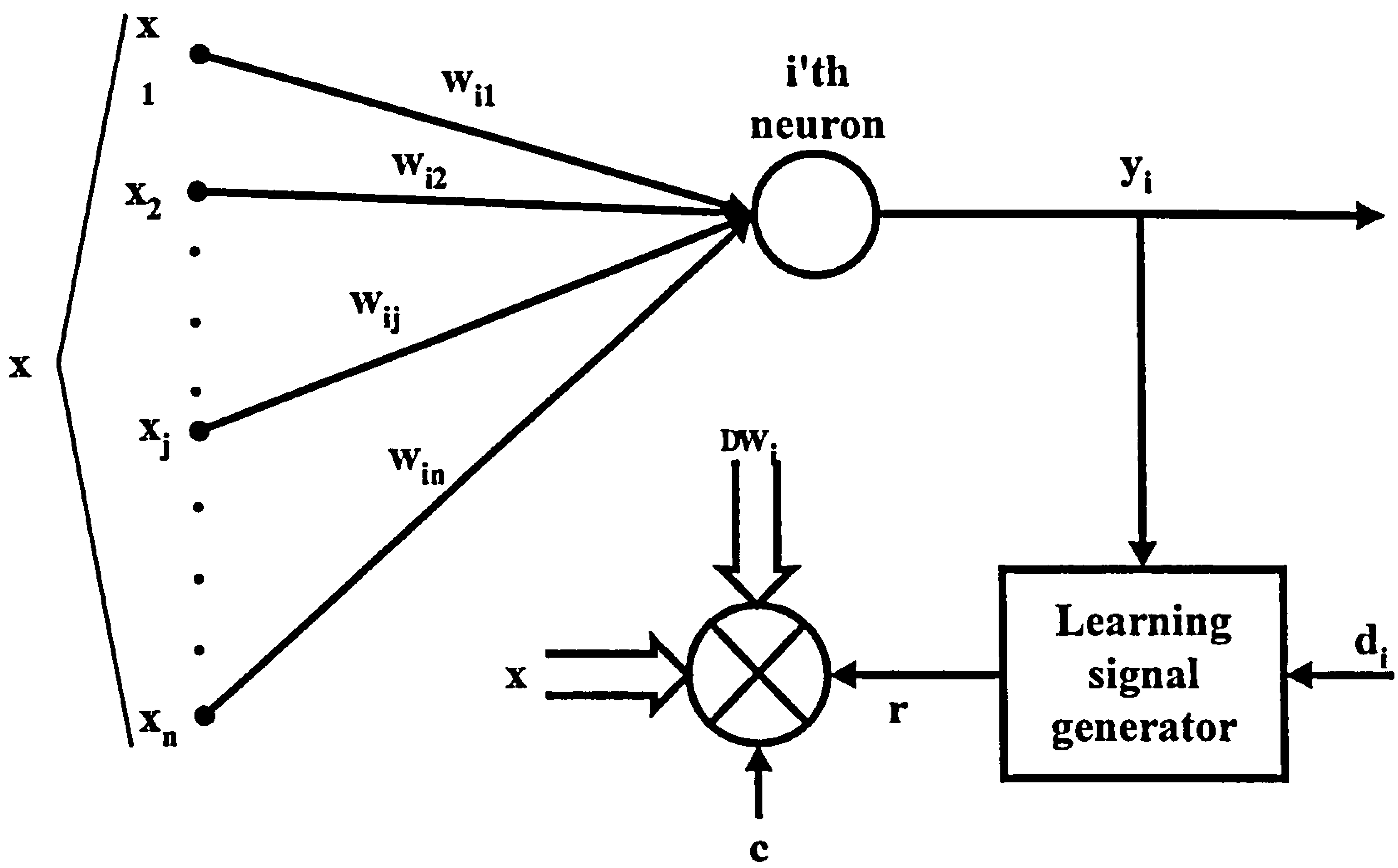

Figure 4.5: The Supervised Learning Method

Equations (4.1) to (4.4) illustrate the supervised learning method

- Learning signal for ith neuron:

$$r = r\,[y_i, d_i] = r[\underline{W}_j, \underline{X}, d_i] \qquad (4.1)$$

- Incremental change in $\underline{W}_i$ (t):

$$\Delta\underline{W}_i\,(t) = cr[\underline{W}_i(t), \underline{X}(t), d_i(t)]\,\underline{X}(t) \qquad (4.2)$$

(where c = learning rate constant)

- Discrete–time learning rule:

$$\underline{W}_i(k+1) = \underline{W}_i\,(k+1) + cr[\underline{W}_i\,(k, \underline{X}(t), d_i(k)]\,\underline{X}(k) \qquad (4.3)$$

- Continuous-time learning rule:

$$d\underline{W}_i(t)/dt = cr[\underline{W}_i\,(k, \underline{X}(t), d_i(k)]\underline{X}(t) \qquad (4.4)$$

4.6.2 Unsupervised Learning

Unsupervised Learning incorporates no external teacher and relies on only local information during the entire learning process. Unsupervised learning organizes data that is presented to the network and discovers its emergent collective properties, examples of unsupervised learning include, Hebbian learning, principal component learning, competitive learning, differential and maximum - minimum learning.

4.6.3 Attributes of Neural Networks

There are several attributes of each Neural Network algorithm but there are six key attributes which are listed in Table 4.3

Table 4.3: Attributes of Neural Networks

Attribute	**Definition**
Train Time	How long does it take the learning technique to capture information adequately?
On-Line/Off-Line	Is the learning procedure an on-line or an off-line procedure?
Supervised/Unsupervised	Is the learning technique a supervised or unsupervised learning procedure?
Linear/Non-Linear	Is the learning technique capable of capturing non-linear mapping?
Structural/Temporal	Does the learning technique capture structural information, temporal information, or both?
Storage Capacity	Is the storage capacity good relative to the number of connections in the network?

4.6.4 Reinforced Learning

Reinforcement learning has a very slow training time, the technique is off-line, supervised and capable of capturing nonlinear mappings. Reinforcement learning captures both structural and temporal information and has a good storage capacity.

Reinforced learning requires one or more neurons at the output layer and a teacher that, unlike supervised learning, does not indicate how close the actual output is to the desired output but whether the actual output is the same with the target output or not. During the learning phase an input stimulus is applied and an output response is obtained (consider one output for simplicity). The teacher does not present the target output to the network, but presents only a 'pass/fail' indication. Thus, the error signal generated during the training session is binary pass or fail.

If the teacher's indication is 'bad', the network readjusts its parameter and tries again and again until it gets its output response right. During this process there is no indication if the output response is moving in the right direction or how close to the correct response it is. Hence, the process of correcting synaptic weights follows a different strategy than the supervised learning process.

Some parameters to watch are the following:-

(i) the time per iteration and the number of iterations per pattern to reach the desired output during the training session

(ii) whether the neural network reaches a global or local minimum, and when in a local pattern if it can get out or if it is trapped

(iii) When reinforcement learning is used as a training technique, certain boundaries should be established so that the trainee should not keep trying to get the correct response ad infinitum.

4.6.5 Competitive learning

It is a form of supervised learning that is distinctive because of its characteristic operation and architecture. In this scheme, several neurons are at the output layer. When an input stimulus is applied, each output neuron competes with each others to produce the closest output signal to the target. This output then becomes the dominant one, and the other outputs cease producing an output signal for that stimulus. For another stimulus, another output neuron becomes the dominant one, and so on. Thus, each output neuron is trained to respond to a different input stimulus. Competitive learning can also be viewed as a random specialization process. When an ANN with competitive learning is part of a greater ANN system, then, because of connectivity issues, this random specialization may not always be desirable. In this case, one might try reinforced learning.

Competitive learning is frequently encountered in groups of people where each member of the group is selected and trained to performed specific tasks based on the principle of the right person at the right time at the right place.

4.6.6 Linear Vector Quantization (LVQ)

The linear vector quantization (LVQ) is a classifier paradigm, developed by Teuvo Kohonen, that adjusts the boundaries between categories to minimize misclassification. An LVQ net has a single layer of nodes (see also winner-takes-all) where each node responds to a class or subclass of patterns. During training, for each input pattern, the LVQ finds the output node with the best match to the training pattern. If the training pattern's class differs from the output node's class, then it finds the next best match. If the input pattern does not produce a better output class then the weight vector will move away from the input. This process, known as competitive learning, in effect moves the boundary between classes closer to the optimum position.

4.7 Network Architecture

The manner in which the neurons of the Neural Network are structured is intimately linked with the learning algorithm used to train the network.

4.7.1 Single-Layered Feedback Network

A layered Neural Network is a network of neurons organized in the form of layers. The simplest layer network has just an 'input' layer of source nodes that are projected onto an 'output' layer of neurons but never vice versa as shown in Figure 4.6 (This network is strictly of a feed-forward type).

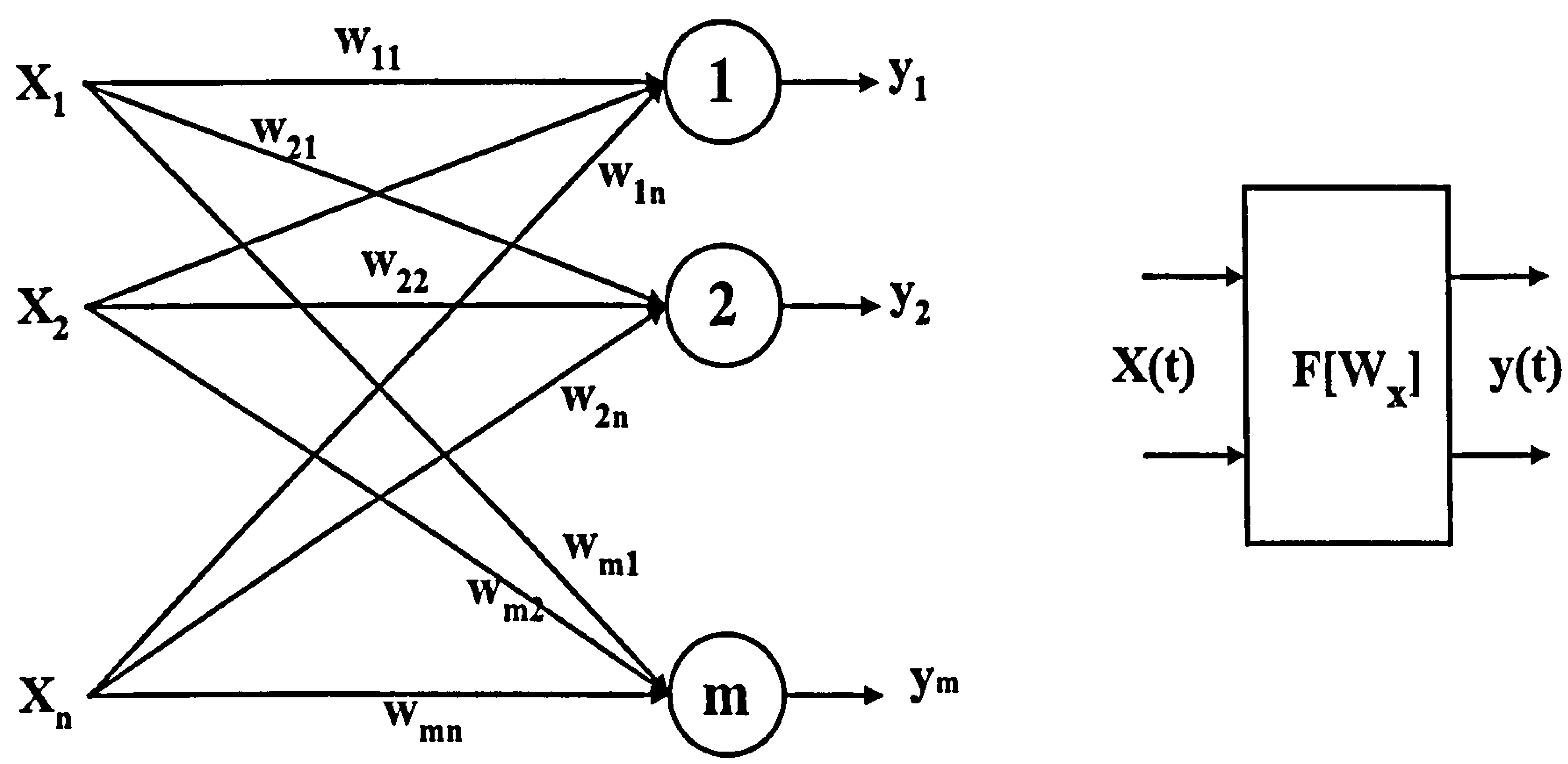

Figure 4.6: Feed-forward Network with a Single Layer of Neurons

Equations (4.5) and (4.6) show the input function X(t) and the output function y(t) respectfully

- Input of the neural network is the n-tuple

$\underline{X}(t) = [X_i \ldots X_n]^T$ (4.5)

- Output of i^{th} neural network

$y_i = f(u_i) = f(\underline{W}_i^T \underline{X}), \quad i = 1,\ldots, m$ (4.6)

where $u_i = \sum w_{ij} X_j, \quad i = 1,\ldots, m$

4.7.2 Single layer FF network (Continuous or binary neurons)

A linear associative memory is an example of a signal-layered Neural Network. In this application, the network associates an output pattern (vector) with an input pattern (vector), and information is stored in the network by modifications made to the synaptic weight of the network.

4.7.3 Multi-layered Feed-forward Networks

The second class of a feed-forward Neural Network distinguishes itself by the presence of more 'hidden layers', whose computation nodes are correspondingly called 'hidden neurons' or 'hidden units'. The function of the hidden neurons is to intervene between the external input and the network output. By adding one or more hidden layers, the network is able to extract high-order statistics. The network acquires a global perspective despite its local connectivity by virtue of the extra set of synaptic connections and the extra dimension of neural interactions. The ability of hidden neurons to extract high-order statistics is particularly valuable when the size of the input layer is large.

The source nodes in the Input layer of the network supply receptive elements of the activation pattern (Input vector), which constitutes the input signal applied to the neurons (computations) nodes in the second layer (i.e. first hidden layer).

The output signals of the second layer are used as input to the third layer and so on, for the rest of the output signals of the preceding layers only. The set of output signals of the neurons in the output (final) layer of the network constitutes the overall response of the network to the activation pattern supplied by the source nodes in the input (first) layer.

$y(t) = f\,[f \ldots f\,[\underline{u}^{(1)}(t)] \ldots]$

$$y(t) = f\,[W(p)f\,[\ldots[w^{(1)}\,\underline{x}(t)] \ldots]] \qquad (4.8)$$

$$\underline{y}(t) = F\{W^{(2)}\,F[W^{(1)}\,\underline{x}(t)]\} \qquad (4.9)$$

Equation (4.8) is a mathematical model of a multi layer feed-forward neural network, Equation (4.9) is a mathematical model of a 3-layered neural network with output.

For a feed-forward network with p source nodes, h1 neurons in the first hidden layer, h neurons in the second layer and q neurons in the output layer, for example, is referred to as a p-h1-h2-q network.

The Neural Network can be said to be 'fully connected' In the sense that every node in each layer of the network is connected to every other node in the adjacent forward layer. If, however, some of the communication links (synaptic connections) are missing from the network, the network is said to be 'partially connected'.

4.8 Comparing Artificial Neural Networks and Other Information Processing Methods

Several informational processing techniques have capabilities similar to those of Neural Network learning algorithms. Some of the alternative methods that are used for pattern recognition, clustering, control and statistical analysis are described below: -

- **Stochastic Approximation** - this was first introduced by Robin and Monro and is a method for finding inputs and outputs when they are extremely noisy. This method is equivalent to back-propagation

- **Kalman Filter** - this is a technique for estimating and predicting the next state of a system based upon a moving average of measurement driven by additive white noise. The Kalman filter requires a model of the relationship between the inputs and the outputs to provide feedback that allows the system to continuously perform its estimation. Back propagation is a special case of Kalman filters

- **Linear and non-linear regression** - this is a technique for fitting a line to a set of data points such that the total distance between the line and the data points is minimized

- **Correlation** - this is a method of comparing two patterns where one pattern is a template and the other, the input. The correlation between the two patterns is a dot product and is used extensively in pattern recognition.

- **Bayes Classification** - the purpose of pattern classification is to determine which class a given pattern belongs. If the class boundaries are not clearly separated and overlap the classification system must find the boundary between the classes that minimize the average mis-classificaflon rate. The smallest possible error is referred to as the Bayes error, and a classifier that provides the Bayes error is called a Bayes Classifier

- **Vector Quaritization** - this method produces a code from a n dimensional input pattern. The code is passed across a channel and then used to reconstruct the original input with random distortion

4.9 The Radial Basis Function Neural Network

This is a function that is symmetric about a given mean (which could be a Gaussian Function). In pattern classification, the Radial Basis Function is used in conjunction with a set of n dimensional reference vectors where each reference vector has a Radial Basis Function that constraints its response. An input pattern is processed through the basis function to produce an output response.

The Radial Basis Function (R.B.F.) is an alternative to the muli-layer network, it adopts a feed forward learning procedure and uses a single internal layer of neurons to learn certain tasks such as classification. The Radial Basis Function is regarded as a special three layered network and has two main advantages over the multi-layer network. Firstly the local representations ensure that only a few units respond to any given input which reduces computational time and secondly the learning rules are linear rather than non-linear, which leads to much faster convergence.

Figure 4.7 shows a schematic diagram of an RBFN with four hidden units.

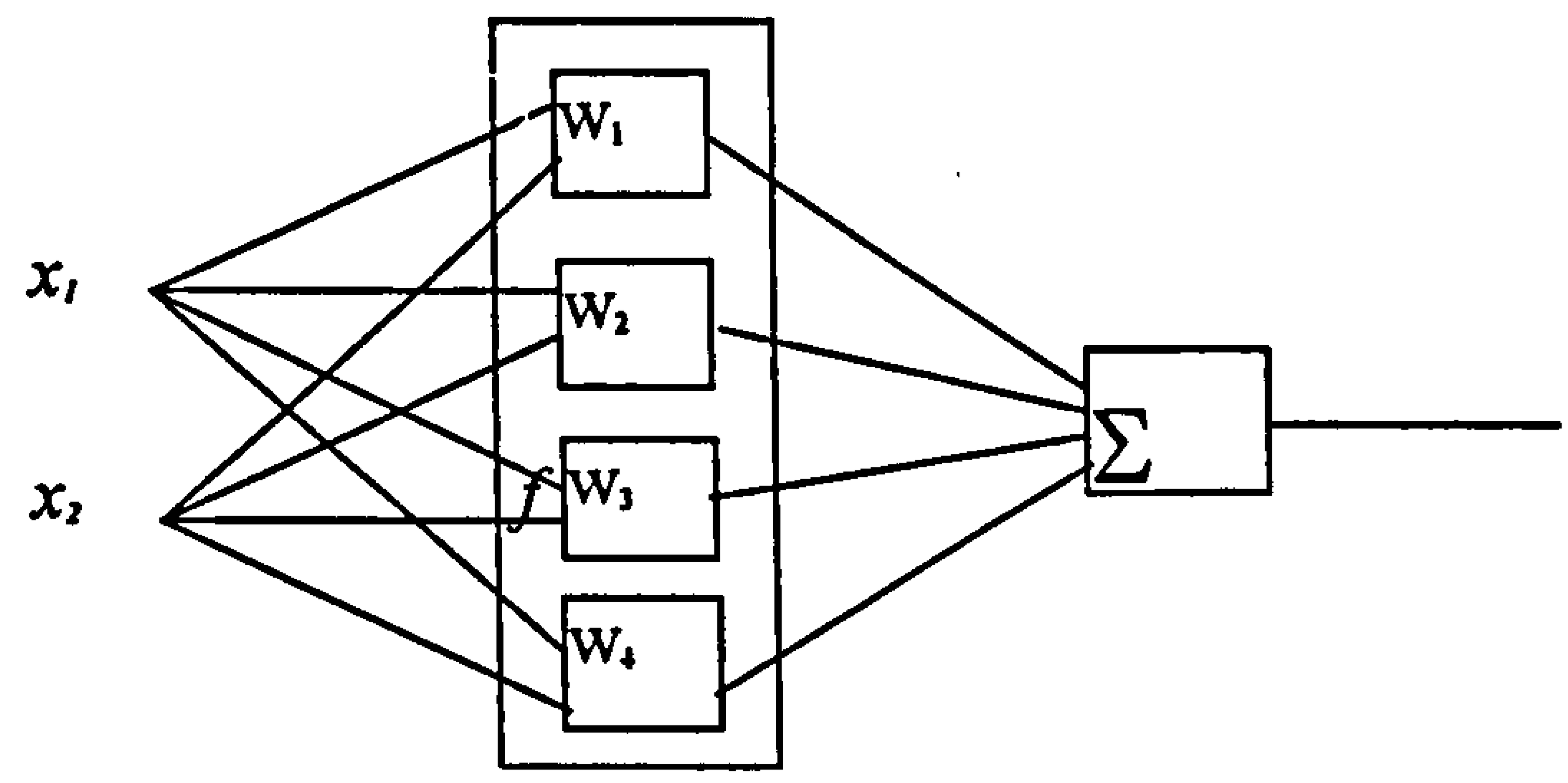

Figure 4.7: A radial basis function network (RBFN).

The activation level of the i-th hidden unit is: -

$$w_i = R_i(\vec{x}) = R_i\left(\|\vec{x} - \vec{c}_i\| / \sigma_i\right),\ i = 1,2,\ldots..n$$

where $\vec{x}$ is a multi-dimensional input vector,
$\vec{c}_i$ is a vector with the same dimension as $\vec{x}$,
n is the number of radial basis function and
$R_i(\cdot)$ is the i-th radial basis function. (4.10)

The output of a radial basis function can be obtained by: -

$$f(\vec{x}) = \sum_{i=1}^{n} f_i w_i = \sum_{i=1}^{n} f_i R_i(\vec{x}), \qquad (4.11)$$

where f_i is the output value associated with i-th layer.

A more complicated method is to calculate weighted average as: -

$$f(\vec{x}) = \frac{\sum_{i=1}^{n} f_i w_i}{\sum_{i=1}^{n} w_i} = \frac{\sum_{i=1}^{n} f_i R_i(\vec{x})}{\sum_{i=1}^{n} R_i(\vec{x})}. \qquad (4.12)$$

In many applications the Radial-Basis Function network offers an alternative to the two-layer network. It can be regarded as a special two-layered network. The network can be efficiently trained to perform tasks via a combination of linear supervised self organizing techniques. The combination of locality of representation and linearity of learning offers tremendous speed advantages relative to back-propagation.

The hidden layer performs a fixed non-linear transformation with no adjustable parameters and it maps the input space onto a new space. The output layer then implements a linear combination on this new space and the only adjustable parameters are the weights of this linear combination. These parameters can be determined using the least squares methods which is an important advantage of this approach. The least squares method provides a simple and efficient means for fitting Radial Basis Function networks. The procedure chooses Radial-Basis Function centres one by one in a rational way until an adequate network has been constructed.

The hidden output weights may be visualized as evolving on a different time scale to the hidden input weights. As the hidden input weight evolve slowly by some non-linear optimization strategy, the output weights adjust themselves rapidly through linear optimization so as to remain In the global minimum of an evolving error surface over the hidden output weights. It can be said the hidden output weights are slaved to the behavior of the input hidden weights.

The construction of a Radial Basis Function Network in its basic form involves three entirely different layers.

- The input layer is made up of source nodes (sensory units)
- The second layer is a hidden layer of high enough dimension, which serves a different purpose from that in a multi-layer perception
- The output layer supplies the response of the network to the activation pattern applied to the input layer

4.10 Recurrent Network

A Recurrent Neural Network (RNN) distinguishes itself from a feed-forward Neural Network in that it has at least one feedback loop.

For example, a RNN may consist of a single layer of neurons with each neuron feeding its output signal back to the Input of all the other neurons as shown in Figure 4.8

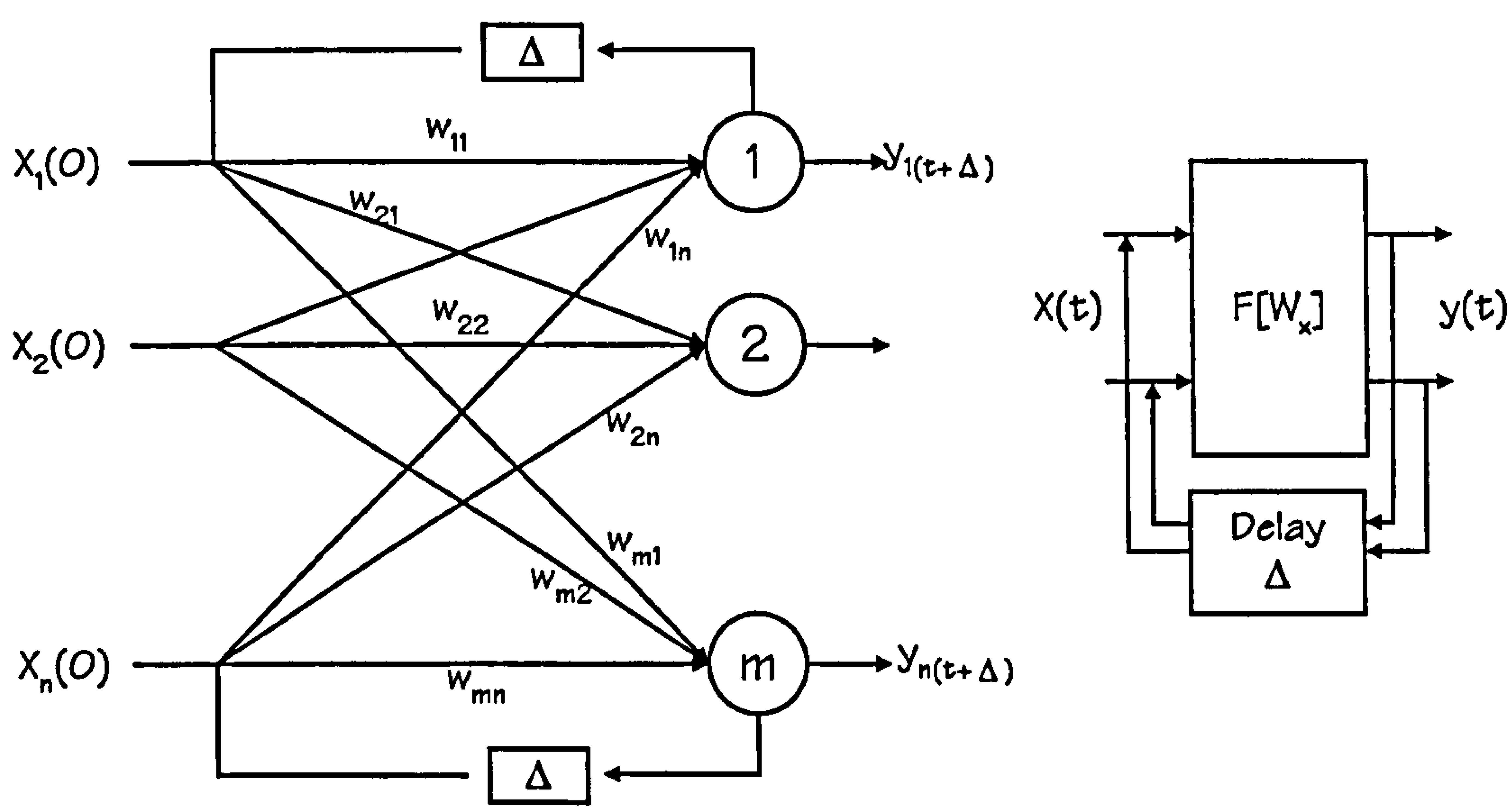

Figure 4.8: Discrete-time recurrent single layer neural network

Input of the neural network is the n-tuple

$\underline{X}(0) = [x_1 \ldots x_n]^T$ (4.13)

Output at time $t + \Delta$

$\underline{y}(t + \Delta) = F[W\,\underline{y}(t)]$ (4.14)

Discrete time neural network output

$\underline{y}(k+1) = F[W\,\underline{y}(k)], \quad k = 1, 2, \ldots$

$\underline{y}(k+1) = F[WF[\ \ldots F[W\,\underline{x}(0)] \ldots\ ]]$ (4.15)

In this structure there is no self-feedback loop in the network, self-feedback refers to a situation where the output of a neuron is fed back to its own input. There are no 'hidden neurons' in Figure 4.8.

4.10.1 Recurrent Network with Hidden Neurons

In this network, the feedback connections originate from the hidden neuron as well as the output neuron. The presence of feedback loops, as in the recurrent structure, has a profound impact on its performance. The feedback loop involves the use of particular branches composed of unit-delay elements, which results in a non-linear dynamical behavior by virtue of the non-linear nature of the neurons. Non-linear dynamics play a key role in the storage function of a recurrent network.

4.11 Self-Organising Map (SOM)

The self-organising map (SOM) is a clustering algorithm developed by Teuvo Kohonen [32]. It creates a map of relationships among input patterns. A SOM network resembles an LVQ network in a number of aspects, both have a single layer of nodes and use a distance matrix to find the output node closest to a given input pattern. Unlike the LVQ, however, SOM output nodes do not correspond to known classes but to unknown clusters that the SOM finds in the data autonomously.

During training, the SOM finds the output node that has the least distance from the training pattern. It then changes the node's weights to increase the similarity to the training pattern, and it influences the weights of the neighbouring nodes even though they have only random relationships to the training pattern. Different patterns trigger different winners that influence different neighbours. The overall effect is to move the output nodes to 'positions' that map the distribution of the training patterns. After training each node's weights model the features that characterise a cluster in the data.

Kohonen's self organising feature map belongs to a class of unsupervised artificial neural network commonly referred to as topographic maps. It serves two purposes, the quantisation and dimensionality reduction of data. The inherent classification properties of the feature map make it a suitable candidate for solving the classification of faults from digital records of Power System faults.

The unsupervised learning approach is used to train winner-take-all units. For each input vector only one such unit will respond, ie. the unit characterised by the maximum output. The units of the network are thus competing for selection. Only the weights of the winner will be adapted. However, this representation is not robust because when one unit is removed (or one cell dies in a biological brain), all information concerning the corresponding class will be lost.

For robust competitive learning Kohonen proposed the *self organisng training algorithm*. Here (ideally) neighbouring neurons classify neighbouring features and thus the loss of a neuron will result in decrease of accuracy but not in a complete loss of information.

4.11.1 Computational Models for Topographic maps

Kohonen [32] proposed a formal model for the formation and function of topographic maps which he called "topology preserving map" and which is known as *Kohonen self-*

organising feature map. For a set of input signals, the map is designed to achieve the following tasks [33]:

(i) Vector quantisation of the input set,

(ii) Dimensionality reduction of the input space,

(iii)Preservation of the topological order present in the similarity relations of the input vectors.

In the following sections the laterally connected architecture, winner-take-all processing, unsupervised learning algorithm and the resulting properties are discussed.

The self organising feature map is an array of m processing elements (neurons) arranged on a lattice of arbitrary dimension. Most applications use a two dimensional lattice, but models where neurons are arranged on a line (one dimensional), or in higher dimensional space can be defined. For a given network the input vectors $\mathbf{x}$ have a fixed dimension n. The n components of the input vectors are connected to each neuron in the lattice. A synaptic weight w_{ij} is defined for a connection from the jth component of the input vector of the ith neuron. Therefore an n-dimensional vector $\mathbf{w}_i$ of synaptic weights is associated with each neuron i.

A neighbourhood relationship is specified between the neurons of the Kohonen network. In the biological cortex, the connectivity of neurons decreases with their relative distance. In the computational model, introducing interactions between neurons whose strength decreases with their distance reproduces this behavior.

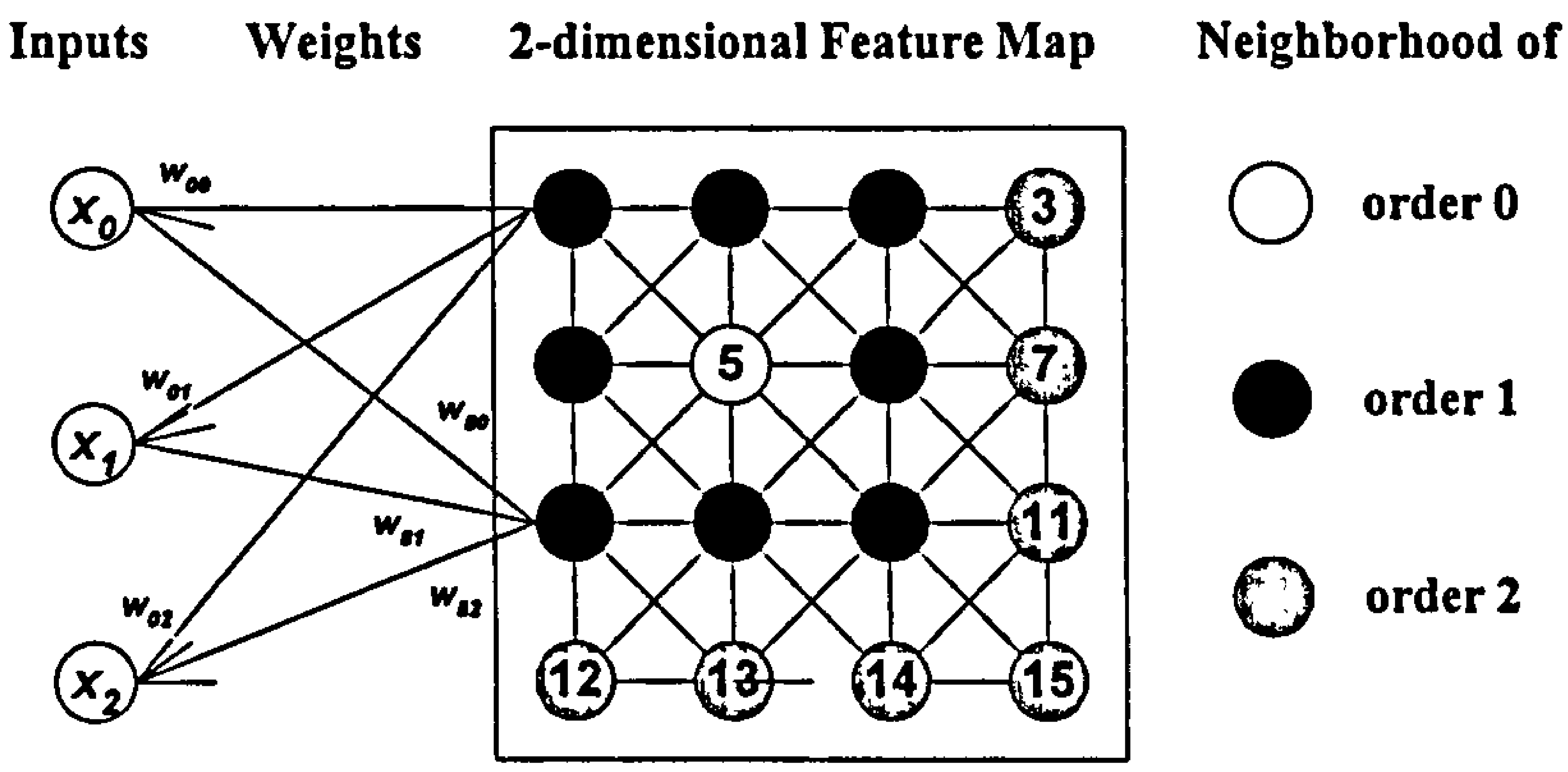

Figure 4.9: A 4x4 network

Figure 4.9 shows an example of a 4x4 Kohonen network which maps 3-dimensional input vectors to a 2-dimensional map containing 16 neurons. Only neurons linked by a black line are connected, and only the links from input components to neurons 0 and 8 and their corresponding weights are represented in this figure.

For the general case of m neurons arranged on a two-dimensional lattice of length m_l and width m_w the connectivity is defined by the following neighbourhood relation.

Each neuron k (k=0, ..., m-1) is associated with its two-dimensional co-ordinate.

r (k): = (ki,kj) for I = 0, ... , mw-1

$$r(k) := (k_i, k_j) \text{ for } i = 0, \ldots, m_l - 1; j = 0, \ldots, m_w - 1 \qquad (4.16)$$

The distance between the neurons k and neuron l is the defined as a function of the indices i and j of k and l,

$$\text{dist}(k,l) = \|r(k) - r(l)\| = \text{floor}(\|(k_i, k_j) - (l_i, l_j)\|) \qquad (4.17)$$

where floor(h) denoted the largest integer less or equal to h. With the Euclidean distance defined in (4.18)

$$\left\|(k_i,k_j)-(l_i,l_j)\right\| := \sqrt{(k_i-l_i)^2+(k_j-l_j)^2} \qquad (4.18)$$

Every neuron in Fig. 4.8 has at most 8 neighborhoods of order one for a distance smaller than 2.

The neighbourhood relationship can be chosen in an arbitrary way, and common ways to connect neurons are the association of four, six or eight neighbours for each neuron.

In principle, the number of neurons is independent of the dimension of the input vector and of the size of the training set. However, a small number of neurons can form only a small number of clusters, each one representing a large set of input vectors, which leads to a coarse discrimination of features in the input vectors. Depending on the application, such a coarse discrimination may not present a sufficiently detailed classification for a large training set and a larger number of neurons may be required.

In biological systems, the lateral connections between neurons implement excitatory and inhibitory links. In his original approach, Kohonen proposed a fully laterally connected network with distance-related strengths of synapses. Neurons close to each other on the grid have a positive (excitatory) coupling, whereas more distant neurons are coupled by negative (inhibitory) connections. These excitatory and inhibitory connections are modelled by two different gaussian functions with different signs.

4.12 conclusion

This chapter has given overview of the many types of Artificial Neural Networks available along with a number of techniques used for their implementation. Although the mathematical concepts are not new, many of them were recorded more than fifty years ago, the introduction of fast computers has enabled many of these concepts to be used for today's complex problems.

The use of Genetic Algorithm based Artificial Neural Networks is demonstrated in chapter six for Electrical Load Forecasting and the use of Self Organising Maps is explored in chapter eight for classifying Power System digital fault records.

Chapter V

NOVEL OPTIMISATION TECHNIQUES

5.1 Introduction

Optimisation is the mathematical process of finding a better, or ideally the best or optimum strategy amongst multiple alternatives that will perform a certain task. Seeking such a strategy has in recent years become more viable for complex engineering systems with the advent of faster and more affordable computing power in the form of everyday personal computers. Of particular relevance to the topic of this documentation is the opportunity to save costs in the area of on-line engineering process control.

The genetic algorithm is a particular optimisation and search method that was first proposed by Holland in the 1970s and has proven itself to be a powerful and general technique [34]. The generality of the method stems from the methods' ability to perform well when little or no domain knowledge is available. It is therefore useful for solving complex engineering system optimisation problems and yet engineers and computer scientists are only just beginning to realise the benefits of such theory [35]. Evolutionary Computation [36] is a name that has been given to describe the field of research that investigates the solution of practical problems by the use of computational processes that simulate natural evolution and genetics.

This chapter gives some background knowledge of the optimisation process and an introduction to the method that is employed in parts of this thesis. In Section 5.2, some general optimisation terminology is discussed. Section 5.3 reviews the field of Evolutionary Computation, including a discussion on the topic of Genetic Algorithms. Other prevalent methods are then reviewed in Section 5.4 and the chapter is concluded with an appraisal of the advantages of the various methods in section 5.5.

5.2 The Optimisation problem

In this section, some basic definitions about optimisation are discussed. This introduction is intended as a prelude to introduction of various optimisation methods in subsequent sections of this chapter.

5.2.1 Definition

References to optimisation or the optimum solution are used frequently throughout this thesis. The Collins English Dictionary defines optimise in the following manner:

" ...to find the best compromise among several often conflicting requirements."

In a more mathematical sense, the following definition can be made [37]

Given a set $D \in \mathbf{R}^n$, known as the feasible region, and given a continuous function $f{:}D \rightarrow R$, the objective function, find

$$\min\{f(x) : x \in D\} \tag{5.1}$$

and the vector $\mathbf{x}^* \in D$ where the minimum is achieved. This point $\mathbf{x}^*$ is then known as the optimal solution to the problem. Optimisation problems can be further classified into non-linear, when function f is not a linear function or linear, which is the case when f is a linear function. The techniques for solving non-linear problems is generally iterative in nature due to the complexity of the functions involved. Problems may also be considered as *combinatorial optimisation* problems if x is constrained to lie in a finite set. Other classifications arise due to the ways of specifying D. For example, D may be defined to be only those vectors that contain only integer numbers. This problem is known as an integer program [37].

It is possible to take the definition of equation (5.1) further to include the concept of local and global optimality [37]. Consider the case when we wish to minimise f(**x**) subject to $\mathbf{x} \in$ D where D is a non-empty subset of $\mathbf{R}^n$. A point $\mathbf{x}^* \in$ D is said to be a *local minimum* if there exists an open set N^1 containing $\mathbf{x}^*$ such that $f(\mathbf{x}^*) \leq f(\mathbf{x})$ for any $\mathbf{x} \in N \cap D$ If $N = R^n$ then $\mathbf{x}^*$ is said to be a *global minimum,* which is the goal of the optimisation search. A problem with only a global optimum is said to be uni-modal, whilst a problem with many optimum points, one of which is a global optimum is said to be multi-modal. It is these problems which are the most difficult to solve Most searches navigating through a multi-modal problem attempt to avoid being trapped in the local minimum state and much research effort is directed towards forcing methods to find the global minimum to a problem. In a particular circumstance, when the feasible region and the objective function are complex, the local minimum is guaranteed to be the global minimum.

5.3 Evolutionary Computation

Evolutionary Computation is the name given to the field of research that investigates simulations of natural evolution. Fogel [36] has subdivided the general term of Evolutionary Computation into two main classifications, Genetic Algorithms and Evolutionary Algorithms, as shown in Figure 5.1. Both are designed as search and optimisation procedures. The distinction has been made to indicate the basic difference between the two approaches in their method of simulating the process of evolving new populations of solutions. In either method, some initial population of possible solutions to the given problem is produced, and evolution towards a solution with the best fitness score takes place by the creation of new populations of solutions formed by operators based on "survival of the fittest" in Darwinian evolution [38]. Evolving individuals and species are viewed as a combination of their genetic programming, the genotype, and their expressed behavioural characteristics, the phenotype [36]. While genetic algorithms operate by altering the genotype in order to drive the evolution process, Evolutionary algorithms emphasise the phenotype by perturbating trial solutions such that there is a near continuous normally distributed variation in the observed behavioural traits of new trials such as would be expected in nature. Evolutionary Algorithms are further classified into Evolutionary Programming, which focus on evolution at the level of reproductive populations, or species, and Evolutionary Strategies, which focus on evolution at the level of individual behaviours. The mechanics of each method is addressed in the following sections, during which obvious and subtle differences are highlighted.

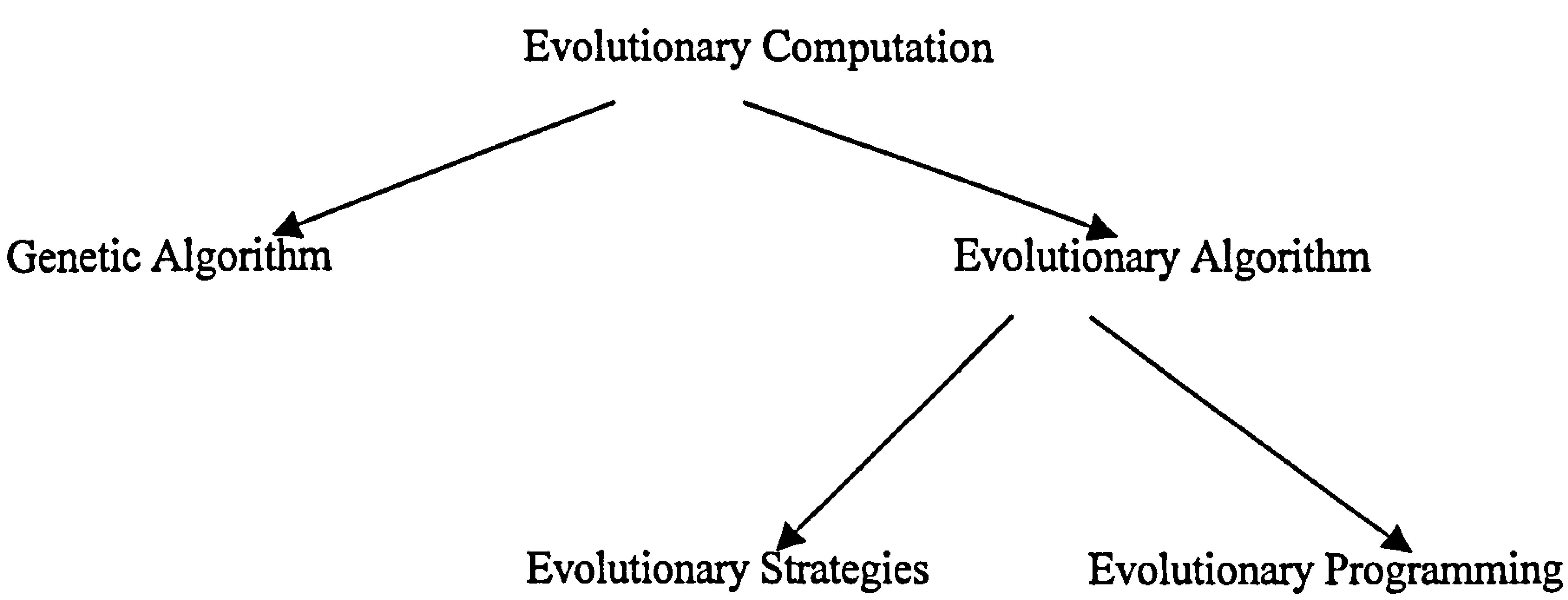

Figure 5.1: Forms of Evolutionary Computation

5.3.1 Genetic Algorithms

Genetic Algorithms (GAs) have been used extensively worldwide, in order to solve various power and other operational control engineering problems. Whilst progress on defining a rigorous theory for the working of Genetic Algorithms has been slow [39], a number of successful applications of the GA to various optimisation problems has demonstrated it's worth. Genetic Algorithms have been used successfully to solve problems in many diverse areas, such as floor plan design [40], VLSI chip placement [41], image restoration in vision systems [42, 43], error recovery in communications [44] and robotics systems [45]. More related to the work in this thesis, optimisation methods based on Genetic Algorithms have been developed to resolve problems in the field of Power Engineering, such as economic dispatch [46, 47], load flow [48, 56] and unit commitment [50, 51]

Once a problem is defined, a suitable representation for possible solutions to the problem in the form of a string of figures must be finalised. The string is sometimes termed a chromosome and is often encoded in binary. A population of possible solutions is then generated. Usually, a set number, s, of chromosomes forms a population. This population is termed the current generation. Solutions strings are given a score which is calculated from the cost function of the problem to be optimised. This function is known as the fitness function. Candidate solutions are selected for participation in the formation of a new population of solutions by a biased random selection process in that candidates with a better fitness function score are more likely to be chosen. In this way, bits of information from the best solutions in the current population are most likely to be passed on to a new population. The method is then driven by two important operators - that of crossover and mutation - which are used to form the new population or next generation of s chromosomes. The two operators correspond loosely to biological counterparts, with crossover analogous to an exchange of genetic material between DNA chromosomes and mutation simulating the appearance of sporadic and random alterations to bits of a string. Once completely formed, this next generation is then termed the current generation and the process is repeated until a desired level of fitness is reached, or a certain preset number of iterations is performed.

Each part of the algorithm will be explained in this section, along with a step-by-step description of how each part is used to form the entire program.

5.3.1.1 Solution Representation and Initial Population Generation

Control problems that require some form of iterative process to find a best solution are generally made up of several, and sometimes conflicting variables. Possible solutions to a problem are represented by a string of values, or chromosome, in the Genetic Algorithm. Most often, values are represented by a binary bit string [47, 48, 39], however values have been represented by floating point numbers [52, 48, 53,54] which are useful for eliminating the discretisation error when continuous variables are being used.

Once the form of the solutions has been designed, an initial population of possible solutions is produced, most often by a random selection method based on predefined constraints, although the initial population may be specified. The use of a population of solutions greatly reduces the chance of a search being trapped into a local minimum. Typically, standard genetic algorithms use a population size of 30 to 200 [39].

5.3.1.2 Fitness Function

The fitness function, f, is a measure of the suitability of a possible solution. This function can be the objective function of the optimisation problem in question, but quite often it is different in some way to aid in computer implementation of the problem. Often in engineering terms, the fitness function is formed in terms of a cost to aid in later definitions. the following items are given.

f_i – The Fitness of candidate solution I, (5.2)

$F\sum_i f_i$ – Fitness sum over the entire population (5.3)

$\bar{f} = \frac{f_i}{F}$ – The normalised fitness of candidate solution i (5.4)

5.3.1.3 Selection of Candidate Solutions

The process of selecting candidate solutions for processing and subsequent entry into a new population is critical to the successful working of the entire algorithm. It is this part of the process that models the "survival of the fittest" concept of evolution. In essence, the strong solutions survive and the weak perish [39]. Selection in this work is achieved by the roulette wheel selection scheme. Each candidate solution in the current generation is allocated a sector of a "roulette wheel" modelled in software form. The normalised fitness of each candidate solution is used to specify the size of the sector, specifically, the angle subtended by the sector at the centre of the wheel equalling $2\pi f_i$. A particular candidate is then chosen if a randomly generated number between 0 and 2π, considered as an angle, falls within the sector allocated to that candidate. This is known as fitness proportionate selection [39]. Note that a candidate may be selected more than once by this method. Other implementations use a model in which certain randomly selected individuals in a sub-group compete and the fittest is selected. This is known as tournament selection and is that seen in nature when males of a species vie for the privilege of mating with a herd of females. Candidates are normally selected in pairs in order to be treated by the crossover operator.

5.3.1.4 The Crossover Operator

The Crossover operator simulates the mating of two candidates to form a pair of child solutions which, if feasible, will be added to the list of solutions in the next generation. Single point crossover is a common method [26] in which, for a chromosome of length l, two child chromosomes are formed from the parents by the swapping of their first m elements, where m is a random number between I and l. Repeated selection and crossover causes continued evolution of the candidate solutions in order to cover a wide area of the solution space and therefore avoid local optimums. However, a trade-off is required to avoid wastage of computation effort in the exploration of possibly less fit areas of the space. When a random number, generated between 0 and 1, is less than a preset value, known as the probability of crossover, crossover is performed. If the random number is greater than the preset probability value, crossover is not applied and the original parents are graduated to the next generation without alteration. In this way, it is hoped that the search will quickly form generations of individuals that survive better in the competitive environment. A commonly used probability figure is 0.6 for a population of 100 [55].

5.3.1.5 The Mutation Operator

If useful genetic information is lost in a population, it cannot be recreated by crossover. When mutation is applied to a chromosome, a randomly selected element is altered if another random number, between 0 and 1, is less than the preset mutation probability [39]. This probability also illustrates another trade-off in the mechanism of the genetic algorithm. If the probability is too high, excessive "new" information could be introduced and possibly valuable information could be destroyed. Alternatively, a low mutation probability would see little new information introduced and the search could be more likely prematurely converge to a local minimum. A commonly used figure for the probability is 0.001 for a population of 100 [55]. The actual mutation itself is dependent upon the solution representation. If the string is binary, the selected element is flipped from 0 to I or I to 0 [39]. For real number implementation, the old value of the element selected is discarded and a new random number within the specified bounds of the element is inserted [49].

5.3.1.6 A Step by Step Algorithm

The procedure followed by the basic genetic algorithm may be summarised as follows:-

1. Generate the initial population of chromosomes either randomly or by some specification. If any candidate is for some reason physically not feasible, it is rejected. This step is complete when the desired number of feasible candidates, s, has been generated. This population is now the current generation.

2. Evaluate the fitness of each candidate by the fitness function. The process may be terminated at this point if:

    a) One or more of the chromosomes have achieved a desired level of fitness (if that level is known)

    b) A homogeneous population is achieved

    c) The maximum number of generations of the process has been reached

3. The next generation of s chromosomes is developed by the following steps

    a) The normalised fitness of each feasible candidate in the current generation is calculated.

b) Select two chromosomes as parents from the current generation by a fitness proportionate selection method such as the roulette wheel scheme.

c) Check for crossover to the selected parent chromosomes the crossover probability. Child chromosomes after crossover should be checked for feasibility. Only feasible child chromosomes are added to the next generation. If crossover is not performed, the original parents graduate to the next generation. Steps b and c are continued until the desired number of feasible candidates, s, has been generated in the next generation.

d) Check for mutation of each chromosome in the next generation probability. If mutation is performed, the new chromosome should be checked for feasibility and discarded from the next generation if it does not meet any physical criteria. Should this occur, procedure b, c, and d should be repeated until a full population, s, in the next generation exists.

4. The next generation formed in step 3 is now termed the current generation and the evolutionary process of the algorithm continues by repeating the steps detailed above starting from step 2.

In the procedure listed above, an iteration of the process occurs each time a new generation has been produced as described by step 3. Often, as described in step 2, a maximum number of iterations is specified and defines the duration of the process.

The Genetic Algorithm's performance is dependent upon the implementation of all of the above steps. One problem that occurs with genetic algorithms is related to maintaining the diversity of the "genetic code" that is held in the strings in the population of solutions. If diversity is lost at an early stage in the process, well before the global solution has been located, the process is less likely to locate that global solution. This phenomenon is known as premature convergence [56]. The following section details some enhancements to the genetic algorithm that have been developed to target this and other problems in the mechanism.

5.3.2 Common Enhancements to the Genetic Algorithm

The following subsections describe some enhancements that can be made to solve different types of problem.

5.3.2.1 Incorporation of Other Techniques

The Genetic Algorithm may be incorporated with other techniques to create a better performing algorithm. Wong and Wong [46] have included a *Simulated Annealing* technique (described in Section 5.4.4) to solve the economic dispatch problem in power engineering. The method is intended to prevent premature convergence and reduce the destructive effects of the mutation operator by using the perturbation techniques and a probabilistic technique for accepting new candidate solutions, both of which are inherent to Simulated Annealing [46].

5.3.2.2 Two Point Crossover

The performance of the genetic algorithm is improved by using an adaptation of the crossover mechanism which has parent solution strings spliced at two points rather than one [56]. The two splicing points on each of the two parent strings are chosen at random and the information between those points is exchanged between the two parent strings to form the child chromosomes for the next generation.

5.3.2.3 Variation of Control Parameters

Srinivas and Patnaik [39] describe the functioning of the genetic algorithm as "... a balanced combination of exploration of new regions in the search space and exploitation of already sampled regions." The balance, they argue, is dependent upon appropriate choice of the control parameters, being the population size and the crossover and mutation rates. However, setting these parameters at static values throughout the life of the algorithm may not always give good results since as the search progresses, the fraction of the search space that is of interest to the algorithm becomes smaller.

An increased crossover probability has the effect of building more new strings, increasing the search rate, but has the drawback of destroying more promising candidates [39]. Researchers have argued [56] that varying the crossover rate will help to prevent premature convergence. Wilson [57] introduced an entropy figure that defined diversity in a population. A falling entropy, indicating less diversity, would increase the crossover rate, whereas a rising entropy would have the opposite effect on the control parameter. The entropy figure is time consuming to calculate and there also exists the possibility that once lost, diversity may not be recovered by moving code from string to string by crossover.

An increase in mutation probability has the effect of turning the method into more of a random search of the solution space, however it also reintroduces a greater amount of diverse genetic information. Maudlin [58] introduced a method termed *uniqueness* whereby the mutation rate was linked to the diversity of the population such that a decrease in diversity caused an increase in the mutation probability. The method allows the genetic algorithm to converge by decreasing the allowed diversity amongst candidates as the number of generations increase. A large amount of computational time, however, is required to calculate the degree of homogeneity amongst the candidates making the method impractical for complex optimisation problems. There also exists the possibility that useful information is not passed onto the parents due to excessive mutation.

An increase of the set population size has the effect of increasing the amount of diversity that is present in the population simply by the number of possible combinations available. An obvious drawback is the computational time required for a search method to sift through a greater number of possible solutions. A choice of population size is therefore dependent upon the problem under consideration.

Research has concluded [39] that optimal static control parameters critically depend upon the nature of the objective function. Indeed, a search for the best parameters form a complex non-linear optimisation problem. However, when testing on a range of objective functions, two distinct parameter sets have appeared as those which are most likely to guarantee good results from a genetic algorithm:

1. A crossover rate of 0.6, a mutation rate of 0.001 and a population size of 100 [59]

2. A crossover rate of 0.9, a mutation rate of 0.01 and a population size of 30 [60]

5.3.2.4 Selection Mechanisms and Scaling

In the initial generations of the genetic algorithm, there typically tends to be a low average fitness value amongst the candidates. If too few strings exist, that have a much greater fitness than others, then the fitness proportionate selection scheme described in Section 5.3.1.3 will allocate a large number of offspring to these solutions, at the expense of others which may have contained the code necessary for the search to locate the global optimum. The result is premature convergence, with the search trapped in a local optimum derived from the early generation "super-strings". At the other extreme, when the search is nearing the final stages of evolution and the difference in fitness amongst the candidates is small, the fitness proportionate scheme is less likely to discriminate the fitter chromosomes and allocate equal numbers of offspring to all candidates. Research has shown that scaling mechanisms and rank based selection schemes overcome these difficulties [39].

Scaling is the practice of changing the fitness values of the population by some predefined function, such as linear scaling where the fitness of each candidate is scaled in the following manner:

$$f_i^1 = af_i + b \tag{5.5}$$

where f_i^1 is the new fitness of candidate i, f_i is it's original fitness value and a and b are suitably chosen constants, which may be recomputed throughout the algorithm if so desired. This may be done in order to avoid negative fitness values, for example.

A different method that is used to avoid the problems discussed in this section is to *rank* the candidates in a generation in order of fitness. The scaled fitness values then vary linearly with the rank of the solution. This introduces a large computational overhead in calculating the ranks of the chromosomes.

Several other methods are also suggested as possible solutions to the premature convergence problem. These include tournament selection and stochastic remainder technique [39]. Both methods have merit as being able to regulate the number of offspring that are spawned from a given solution.

5.3.2.5 Elitism

When a number of the fittest individuals in a given generation are retained in the next generation, a standard of fitness amongst possible solutions is retained from generation to generation. In this way, new generations are guaranteed to contain members that are at least as fit as those of the previous generation. Such a practice is termed *elitism* [39] and the algorithm is classed as a *steady-state G.A.* Algorithms that replace the entire population are known as generational G.A.s. Since the level of fitness is guaranteed for the steady-state G.A.s, they tend to be characterised by high population numbers and large rates of crossover and mutation amongst the low fitness members of the population.

5.3.2.6 Variation of Encoding

As mentioned in Section 5.3.1.1, binary encoding of solutions for use by the genetic algorithm procedure have traditionally been used due to their ease with which crossover and mutation routines may be implemented. When schemes other that the binary {0,1} set are used, the mutation and crossover operators must be adjusted to suit that implementation. One particular example of an encoding method that avoids the discretisation effect of binary encoding on continuous variables is the floating point number coding method developed by Wong and Wong [53]. This implementation fully revises the crossover and mutation procedures to suit the implementation.

5.3.2.7 Distributed and Parallel Genetic Algorithms

Distributed and parallel algorithms are implemented in an attempt to improve the efficiency of the genetic search by utilising new technology in parallel computing. The aim is to simultaneously evolve strings in independent implementations [39]. The two approaches differ in that distributed Gas encompass a number of weakly interacting sub-populations that each carry out a search on the same domain, whereas independent parallel implementations of the same algorithm on several different machines forms the basis for parallel GAs.

In the case of the distributed genetic algorithm, a large population is subdivided into the smaller groups which evolve independently. The best strings from the various populations are exchanged in order to maintain global competition throughout the algorithm. Because of the independent nature of the sub-populations, the algorithm may be implemented on connected parallel engines or on a single machine. Parallel GAs however, have been designed specifically for implementation on separate parallel machines. Physical issues of parallel computation, such as global communication and synchronisation have forced some changes to the structure of the algorithm [61].

5.3.3 Evolutionary Programming

Evolutionary programming (EP) [62] was originally conceived by Lawrence J Fogel in the early 1960s [84]. As with Genetic Algorithms, evolutionary programming is a stochastic optimisation strategy but unlike the GA which emulate specific genetic operators, emphasis is placed upon the behavioral links between the parents and their offspring [36]. A trial solution to a particular problem is described by a vector x. The basic EP method then involves 3 steps:

1) An initial population of P trial vectors, or parent vectors, $\mathbf{x}_i$, l = 1,...,P, is selected at random from a feasible range of values for each of the dimensions of the vector.

2) An offspring vector, x_i, i = 1,...,P, is created from each parent by adding a zero mean Gaussian random variable with preselected standard deviation to each dimension of x. This is regarded as a mutation process.

3) Each offspring solution is assessed by computing its fitness by use of a fitness function, $F(x)=R^n \rightarrow R$. Selection then determines which of these vectors to maintain by comparing the values $F(x_i)$ and $F(x'_i)$, i = 1,...,P. If the process is a maximisation, then the P vectors that possess the greater values of F become the new parents for the next generation. Note that there is no requirement that the population size remain constant, nor that only a single offspring be generated from each parent.

The process of generating new trials and selecting those with the best fitness continues until a sufficient solution is reached or the available computation time has expired. Note that evolutionary programming does not use any form of crossover as a genetic operator hence there is no sexual recombination. The solutions are therefore considered to *be abstracted species*. It is assumed that whatever genetic transformation has occurred, the resulting change in each bit of the solution vector will follow a Gaussian distribution with zero mean and some standard deviation. Each bit is therefore considered to be a "behavioural trait" of the species, or the phenotype [36]. Since specific genetic alterations can affect many behavioural characteristics, it is appropriate to simultaneously vary all of the components of a parent in the creation of a new offspring.

Evolutionary Programming which focuses upon a single parent, single offspring search is termed a (1+1)-EP. Extensions to multiple parents and offspring are described as (μ+λ)-EP and (μ,λ)-EP. For the former, μ parents are used to create λ offspring and the best μ

solutions from all $\mu+\lambda$ are selected as new parents. In the latter, the new parents are selected only from the λ offspring.

5.3.4 Evolutionary Strategies

Evolutionary Strategies were conceived in 1963, independently of the evolutionary programming technique, by Ingo Rechenberg, Hans-Paul Schwefel and Peter Bienert [64 - 66], three students at the Technical University of Berlin. They were searching for the optimal shapes of bodies in a flow produced by a wind tunnel. The original structure of the evolutionary strategy at this time was a simple two-membered or (1+1)-ES, where one parent generates one offspring per generation by applying normally distributed mutations, similar to the evolutionary programming method, until a child performs better than its ancestor and takes its place. In this simple case, the best ratio between successful and all mutations was noted to be 1/S. This is also known as the 1/S success rule [62]. Later expansions of the procedure introduced the use of multiple parents and offspring with two distinct approaches, $(\mu+\lambda)$-ES and (μ,λ)-ES optimisation which followed the guidelines detailed for their counterparts in the evolutionary programming technique described in Section 5.3.3. The second scheme is considered to be more closely related to natural evolution.

The $(\mu+\lambda)$-ES or (μ,λ)-ES strategies incorporates information at the genotype or genetic code level, which is passed from the parents to the child solutions [67], thus effectively classifying each candidate as an individual entity. It is the existence of this "genetic" information which sets evolutionary strategies apart from the evolutionary programming technique. The latter regards the candidates as abstracted species since evolution is strictly related to behavioural, or phenotypic changes from parent to child candidates.

A single individual of an Evolutionary Strategy population consists of the following items in its, genotype [36]:

i. The object variables are a set of real-number values, x, that form a vector, $\mathbf{x}_i$, i = l,...,P, where P is the population size. These values are those that must be tuned such that the objective or fitness function will reach its global optimum.

ii. The strategy variables, σ, represent the standard deviation of a $(0,\sigma)$ Gaussian distribution (GD) being added to each object variable as a mutation. These also form a vector, σ_i, I = l,...,P.

With an "expectancy value" of 0 for the Gaussian variable that is added to object variables upon mutation, parents will, on average, produce a child that is similar to themselves. The strategy variables, however, mutate from generation to generation in a log-normally distributed fashion, that is, exp(GD). By doing this, an object variable that has been successfully mutated by a large step in a past generation, will be more likely to mutate by a large step in the next generation. This reflects a self-adaptation by the population of the step sizes between generations, which models the way in which selection prefers those individuals that have built a better model of the objective function, thus producing better offspring. This self-adaptive information is purely at the level of the genotype, or the genetic code [83]. Learning in ES therefore takes place on two levels:

i. At the genotypic level by alteration of the object and strategy variables.

ii. At the phenotypic level by alteration of the fitness or objective function, f(**x**), where **x** denotes the vector of objective variables

Further research by Schwefel [68] has shown that the (μ,λ)-ES approach, which allows for intermediate deterioration of solutions, performs better than the (μ+λ)-ES approach He argues that only by neglecting highly fit individuals can a permanent adaptation of the step sizes take place, thus avoiding long phases of stagnation due to mis-adapted strategy variables. By choosing a certain ratio of μ/λ the convergence property of the Evolutionary Strategy may be determined. For a fast but local convergence, a small or hard selection ratio is appropriate, for example 5/100, but when searching for a global optimum, a softer selection is more suitable (eg. 15/100) [68].

5.4 Other Common Optimisation Methods

This section is intended to summarise some other more traditional optimisation and search methods that have been popular for solving difficult engineering problems.

5.4.1 Lagrange Multipliers

Many optimisation problems may seem on the surface to be easily solved if they are described by a set of simultaneous equations. In most cases, there exist a variety of constraints on participating variables which make the optimisation task difficult. The Lagrange multiplier method of optimisation attempts to deal with these constraints whilst solving the optimisation by the direct use of the simultaneous equations which describe the problem.

5.4.1.1 Origins of the Method

Consider the problem of minimising a function **f** in **n** variables where each component of the function may be expressed in terms of a quadratic equation of some variable **x**.

$$f = \sum_{i=1}^{n} f_i = \sum_{i=n}^{n} \left(a_i + b_i x_i + c_i x_i^2\right) \qquad (5.6)$$

a_i, b_i and **c_i** are constants for **i=1,...,n**

If there are no constraints on this problem, then the optimal value for **f** may be found from:

$$\frac{\partial f}{\partial x_i} = 0 \qquad (5.7)$$

Which would imply that

$$x_{i,\mathrm{optimail}} = \frac{-b_i}{2c_i} \qquad (5.8)$$

Such examples are rare in practice due to the existence of various constraints on a problem. Suppose in the above example, there exists a constraint that variables x; are required to sum to a particular value, S:

$$S = \sum_{i=1}^{n} x_i \qquad (5.9)$$

There is now an extra equation to be considered and simple use of partial derivatives such as those of equation 5.7 is no longer valid due to the interdependence of the variables $\mathbf{x_i}$. It is possible to include the constraint by eliminating one variable, $\mathbf{x_m}$, in the following manner:

$$x_m = S - \sum_{i=1, i \neq m}^{n} x_i \qquad (5.10)$$

When substituting this equation into (5.6), the following results:

$$f=\sum_{i=1,i\neq m}^{n}\left(a_i+b_ix_i+c_ix_i^2\right)+a_m+b_b\left(S-\sum_{i=1,i\neq m}^{m}x_i\right)+c_m\left(S-\sum_{i=1,i\neq m}^{m}x_i\right)^2 \quad (5.11)$$

The objective function, **f** has been transformed to a function in (**n**-1) variables with all constrained equations included and the optimal solutions may now be found by the method used for the unconstrained case. This simple method will, however, run into difficulties when constraints are non-linear and variable elimination is therefore not so straight forward. The Lagrange multiplier technique [90] may be used in this case.

5.4.1.2 Optimisation Solution by Lagrange Multipliers

In the general case, the optimisation problem discussed in Section 5.2.1 may be expanded to include m constraint conditions:

Minimise f(**x**) such that

$$g_j(\mathbf{x}) = 0 \text{ for } j= 1,\ldots,m_e \quad (5.12a)$$

$$\text{and } g_j(\mathbf{x}) \geq 0 \text{ for } j= m_e+1,\ldots,m \quad (5.12b)$$

where **x** is a vector in R^n and the objective function **f** and constraint equations $g_1,\ldots,g_m$ are assumed to be twice differentiable. The constraint equations, (5.12a) and (5.12b) form the feasible region $D \in R_n$ as defined in Section 5.2. 1.

In order for a vector **x*** to be a local solution of the constrained optimisation problem, it must be a feasible point. Therefore it must meet equation (5.12a), known as the equality conditions and also equation (5.12b) known as the inequality conditions. Some of the inequality constraints will be met at the boundary of the feasible region such that $g_j(\mathbf{x}^*)$ will be exactly zero for those constraints. Any constraints are considered to be binding constraints if:

$$g_j(\mathbf{x^*})=0 \tag{5.12c}$$

If the vector $\mathbf{x^*}$ is to be a local *minimum*, then the objective function, $\mathbf{f(x^*)} \leq \mathbf{f(x)}$ for all feasible points $\mathbf{x}$ near $\mathbf{x^*}$. As $\mathbf{x^*}$ meets the constraint functions of (5.12), the gradient vector $\nabla\mathbf{f(x^*)}$ will lie in the space spanned by the normal functions of the constraints, $\nabla g_j(\mathbf{x^*})$, hence there exists scalars $\lambda^*_1,\ldots,\lambda^*_m$ that will form the equality:

$$\nabla f\left(x^*\right)-\sum_{j=1}^{m}\lambda_j^*\nabla g_j\left(x^*\right)=0 \tag{5.12}$$

The scalars $\lambda^*_1,\ldots,\lambda^*_m$ are named the *Lagrange multipliers* [69]. The function

$$L^*(x)=f(x)-\sum_{j=1}^{m}\lambda_j^* g_j(x) \tag{5.14}$$

is termed the *Lagrangian* function and therefore equation (5.13) may be expressed in the form

$$\nabla L^*\left(x^*\right)=0 \tag{5.15}$$

It is important to note that the Lagrange multipliers of the inequality constraint functions will satisfy [90]

$$\lambda_j^* \geq 0 \tag{5.16a}$$

and $\lambda_j^* g_j\left(x^*\right)=0$ for $j=m_e+1,\ldots,m$ (5.16b)

Therefore for any non-binding inequality constraints, $\lambda^*_j = 0$.

The method for actually solving for **x*** involves a numerical iterative process in which initial values for x are produced and forced to converge to **x*** and the associated scalars λ^*_j by using equations (5.12) and (5.13) as measures to indicate when the process may be terminated. To formally state this process, define the term **T(x,λ)** where

$$T(x,\lambda) = \|\nabla L\| + \|v\| \tag{5.17}$$

In this instance, **L** is the Lagrangian function found from the estimated **x** and **λ** values and v is a vector of violations of the constraints [69]. That is

$$L(x) = f(x) - \sum_{j=1}^{m} \lambda_j g_j(x) \tag{5.18}$$

$$v_j = g_j(x) \qquad \text{for } j=1,\ldots,m_e \tag{5.19a}$$

$$= \min(0, g_j(x)) \quad \text{for } j= m_e+1,\ldots,m \tag{5.19b}$$

If all optimality conditions hold then an iterative scheme for solving the optimisation problem defined by equation (5.12) would have converged if **T(x,λ)** as defined by equation (5.17) was sufficiently small. One of the inherent problems with the Lagrange multiplier method is that in the situation when there are a large number of inequality and equality constraints, the special measures in the solution process mentioned above can cause difficulties when attempting to determine solution schedules in combinatorial optimisation problems [70].

5.4.2 Dynamic Programming

The formalisation of the Dynamic Programming approach to optimisation was provided by Bellman [71] in 1954. The technique is most useful when a problem can be described by a series of single decisions made one after the other. It solves a problem by reducing it to several sub-problems and then combining the solutions to these in order to form an overall solution. In the situation where the sub-problems are not independent, that is to say that the sub-problems share further sub-problems, the dynamic programming technique is particularly useful since it solves each 'sub' sub-problem only once and saves the solution to memory, thereby avoiding the need for time consuming re-computation for each sub-problem.

5.4.2.1 Theoretical Origins

The optimisation definition of equation (5.1) is effectively a problem of simultaneously finding n variables, as $\mathbf{x}^*$ is a member of R^n, The Dynamic programming method would seek to solve a succession of n problems, or stages, of solving for one variable. It is often useful to consider the problems to be decisions made sequentially in time for ease of understanding. After each decision is made, the system under consideration would have changed it's particular characteristics or state before the next decision is to be made. Since it would not be known how the other n-1 decisions would have changed the state of the system, it is necessary to calculate the best decision value for a given variable over many possible states of a system. The system itself will pass through a progression of states with each decision variable being calculated and the sequence of the n decisions would yield a value for the original function which is being optimised. Dynamic programming is therefore an exercise in trial and error, with all possible sequences of decisions being compared for the optimum solution. A global optimum is guaranteed but for a complex system of many variables, the computational time for calculating such a solution can be immense due to the combinatorial explosion.

5.4.3 Local Search

The local search method is so called because of its' foundations in the concept of local optimality. In optimisation problems, it guarantees to find a local optimum solution to a problem.

The basic mechanics of the method are simple. Consider a combinatorial optimisation problem in which some solution space or feasible region may be recognised. To maintain consistency with the definition of Section 5.2.1, let D be that space and let $\mathbf{x}^*$ be a solution state in D. In addition, define E as a set of feasible solutions within some neighbourhood of $\mathbf{x}^*$. When the cost function associated with $\mathbf{x}^*$ is shown to be less than or equal to any associated with any other solution point in E, then $\mathbf{x}^*$ is said to be the local optimum (minimum) with respect to the region E Based upon this premise, the local search method simply works upon a number of solutions in a space and checks the local region until a predefined precedent has been reached. The actual mechanism by which this method is implemented is more complex.

Consider a situation when a feasible solution vector, $x \in R^n$, to a given problem is known, but it is not known if it is the optimum solution. A solution in the neighbourhood of the values in $\mathbf{x}$ is found by perturbing the values in $\mathbf{x}$ according to some probability distribution function (PDF). Often the Gaussian PDF with a zero mean and a preset standard deviation. The addition of perturbations to each value of the vector $\mathbf{x}$ forms a new vector $\mathbf{y}$, which, once deemed or altered to be feasible, is the new solution within a neighbourhood of the original solution $\mathbf{x}$. If the objective function of the optimisation problem under consideration proves that the vector y is a better solution to the problem than the original vector **x**, then vector **y** is taken as the current solution for generation of another neighbourhood solution, otherwise vector **x** is retained.

An example presented by Wong and Fung [72] specified a local search technique for the economic dispatch problem in power engineering. This solution specifies that the local search process proceeded in iterations. Within each iteration, neighbourhood solutions are generated one by one, with each newly generated feasible neighbourhood vector competing for the right to generate the next vector with it's progenitor. When a specified number of solutions has been created, the last accepted solution is the starting point for the next iteration. To further enhance the process, the standard deviation of the Gaussian PDF used to create perturbations is altered according to the formula

$$\text{Standard Deviation} = \gamma\sigma_k, \qquad \sigma_k = r^{(k-1)}\sigma_1 \tag{5.31}$$

where γ is a preset constant, r is a preset constant less than one, σ_1, is a predefined initial value and k is the iteration number.

Use of this enhancement reduces the range of the neighbourhood space of the current solution as the number of iterations increases. The entire process halts when the number of iterations has exceeded a predefined number, or when there is no improvement of the solution over a certain number of iterations, also predefined.

It is interesting to note that the local search method is similar to the (1+1)-EP described in Section 5.3.3, however there does not appear to be any suggestions to extend the method to a $(\mu+\lambda)$ or a (μ,λ) strategy as covered by the literature in the development of the evolutionary programming method. The local search method forms some of the foundations for the popular *simulated annealing* method of optimisation discussed in the next section.

As previously stated, the local search method guarantees a local optimum solution. It will only find a global solution if the objective function of the optimisation problem is uni-modal. If this is not the case, then the chance of the local search method happening upon the global solution is enhanced if

1. The neighbourhood of the solution space is enlarged
2. Several executions are compared with a variety of initial solution vectors
3. Assistance is given to the algorithm to periodically avoid entrapment in a local solution by allowing for the acceptance of a worse solution at random points in the process.

The third option presented here is inherent to the simulated annealing process.

5.4.4 Simulated Annealing

The simulated annealing [70] method of optimisation is based upon the process which is seen in the physical annealing process of solids.

5.4.4.1 Origins of the Method

When physical solids are cooled from a high temperature, molten particles can acquire a state of thermal equilibrium at any temperature. The thermal characteristics of such a state are characterised by a distribution known as the Boltzmann probability factor (BPF).

$$\mathrm{BPF} = \exp\left(\frac{-\mathrm{E}}{\mathrm{K_B T}}\right) \tag{5.32}$$

where E is the energy of the configuration of the particle, K_B is a constant known as the Boltzmann constant and T is the temperature.

The probability that the particle would be in a state such that it possessed energy E_i, $P(E_i)$ is therefore given by

$$P(E_i) = \frac{\mathrm{BPF} = \exp\left(\frac{-E_i}{K_B T}\right)}{\sum_j \mathrm{BPF} = \sum_j \exp\left(\frac{-E_j}{K_B T}\right)} \tag{5.33}$$

where the denominator is the summation of all BPFs of the possible states that a particle can have at temperature T.

This computation of the Boltzmann probability factor for all feasible states of the particle is analogous to the generation of a number of neighbourhood solutions as was proposed for the local search algorithm detailed in Section 5.4.5. Here, a newly generated energy state is conditionally accepted as the state of the particle in preparation for the next decrease in temperature. In addition, for the simulated annealing method, the acceptance criteria for the adoption of a newly generated solution is somewhat different.

When considering the state that a molten particle will assume for a given temperature, the follow criteria are considered.

1. That a particle will assume the state with the lowest energy, or

2. That a particle will, with a certain probability, assume a state with a higher energy.

For a given temperature, the molten particles may therefore settle into a state of energy that may not be considered optimum, which would correspond to being at the lowest energy level. However with a drop in temperature, the particles again have the opportunity to settle at a new level of energy until finally at a low enough temperature, the material solidifies.

This search for an energy level forms the basis for the use of simulating the annealing process in order to solve an optimisation problem. A "particle" may be a solution vector to the problem, it's values being alterable so as to find the best "energy level" at which it may reside, or in terms of the optimisation problem, the best solution to the objective function. The probability that a less than optimum solution may be accepted at a given "temperature" gives the simulated annealing process the ability to avoid local optimums and "jump" to possibly a better solution at a lower "temperature" that perhaps may not have been attainable at the optimum solution for a higher "temperature".

5.4.4.2 Application as an Optimisation Technique

In order to use the simulated annealing technique to solve a mathematical optimisation problem, various properties of the cooling process are adopted.

Firstly, the temperature variable may become a control parameter for the process. In the application by Wong and Fung [72], the cooling scheduled adopted was characterised by the equation

$$T_k = r^{(k-1)} \times T_1 \tag{5.33}$$

where k is the cooling step counter, r is a scaling constant less than one and T is the initial temperature.

This schedule is very similar in nature to that of equation (5.31) for the local search method

With the temperature figure defined for a given step in the cooling process, the probability that a non optimum solution, or "energy state" is selected may be defined in the application presented by Wong and Fung [72], the probability that an particle state with a higher energy be accepted was

$$P(\Delta) = 1/(1 + esp(\Delta/T)) \qquad (5.34)$$

where Δ is the change in the energy between the current state of the "particle" and a state formed by small perturbations of the current state and T is the temperature at that stage of the cooling process.

With these parameters defined, a schedule for the optimisation procedure may be detailed. The process begins at some predefined initial temperature.

1. A feasible solution vector to an optimisation problem is generated. This may be done by some random selection process. The value of the objective function for this solution is calculated. This initial vector is the current solution.

2. Values in the solution vector are perturbed to form a new solution vector. This perturbation may be performed randomly, however in some cases [70] it may be necessary to control the amount of perturbation to ensure that a feasible solution to the problem may be easily generated.

3. The cost or objective function of the new vector is compared with that of the original vector. If the new vector is a better solution to the optimisation problem as specified by the value of the objective function, then the new vector is now the current solution.

4. If the vector found in step 1 is a better solution to the problem, then the new vector created in step 2 is accepted as the current vector with a probability given by equation (5.34). If the new solution is not adopted, then the vector found in step 1 is retained as the current solution.

5. After a set number of new solutions have been generated, the temperature is lowered in accordance with equation (5.33) and the most recent current solution is retained for this new iteration. The process returns to step 2 until a maximum number of iterations is reached, or there is no significant improvement in the objective function after a preset number of iterations. The optimum process is then terminated

The simulated annealing process has proved to be a popular optimisation tool.

5.5 Conclusion

This chapter has stated a definition of optimisation and has discussed various types of such problems. It has then reviewed the Evolutionary Computing family of optimisation methods with particular emphasis on the Genetic Algorithm, which is used in investigation within this thesis. Various other popular methods of optimisation were then reviewed.

Possible solutions to optimisation problems were introduced to be either local or global minimum solutions with the latter being the desirable result. The evolutionary computation, which has potential to produce a global solution to a problem due to the searching mechanisms which are inherent to the procedures. Various mechanisms may be introduced to the genetic algorithm routine which may eliminate the problems of premature convergence, thus enhancing the methods' chances of producing the best solution. The other, more traditional methods of optimisation described in this chapter included Lagrange multipliers, Dynamic Programming, Local Search and Simulated annealing. Only the Dynamic Programming method guarantees a global optimum solution to an optimisation problem, however for complex problems, the method could take a vast amount of time to locate a solution due to the potential for combinatorial explosion since every possible solution is considered. The Lagrange multiplier method and the local search method are useful for quick location of a global minimum and are therefore useful when the topography of the optimisation problem is uni-modal. However in a complex multi-modal problem, a global solution is less likely. The simulated annealing method has been more popular for solving complex multi-modal problems since it includes techniques for the search to avoid being trapped in local minimum solutions.

Chapter VI

GENETIC ALGORITHM BASED ARTIFICIAL NEURAL NETWORKS FOR ELECTRIC LOAD FORECASTING

6.1 Introduction

An accurate and stable load forecast is essential for many operating decisions taken by Utilities. The short-term load forecast provides the information to be adopted in the on-line scheduling and security functions of the energy management system, such as unit commitment, economic dispatch and load management. Hence, accurate load forecasting is essential for the optimal planning and operation of large-scale power systems.

Many techniques have been proposed and used for short-term load forecasting. Time-series models based on extrapolation are used for the representation of load behavior by trend curves. The time series approach, regression approach, state-space models, pattern recognition and expert systems are also some of the other techniques used [73-77].

The time series approach assumes that the load at a point in time depends mainly on previous load patterns, such as the auto-regressive moving average models and the spectral expansion technique [74]. The regression method utilises the tendency that the load pattern has a strong correlation to the weather pattern. The weather-sensitive portion of the load is arbitrarily extracted and modeled by a pre-determined functional relationship with weather variables. All the above approaches use a large number of complex equations that involve lots of computational time. More recently, artificial neural network (ANN) techniques have been used in many modeling problems [78-81]. One of the most popular training algorithms for feed-forward ANNs is a gradient descent search algorithm, for example, the back-propagation (BP) approach, which tries to minimise the total Mean Square Error (MSE) between actual output and target output of an ANN. This error is used to guide BP's search in the weight space. There have been some successful applications of BP algorithms in various areas. However, drawbacks with the BP algorithm do exist due to its gradient descent nature. It often gets trapped in a local minimum of the error function and is very inefficient in searching for a global minimum of a function which is vast, multimodal, and non-differentiable. One way to overcome BP's as well as other gradient descent search-based training algorithms' shortcomings, is to consider the training process as the evolution of connection weights towards an optimal (near optimal) set defined by a fitness function. From such a point of view, global search procedures like GAs can be used effectively to train an ANN. Therefore, GA integrated with ANN (GA_ANN) has been implemented for searching a solution. Object Oriented Techniques (OOT) were the framework for integrating ANN and GA. OOT gives us the ability to combine the existing objects (ANN object and GA object) and create new components (GA_ANN).

6.2 Genetic Algorithm and Artificial Neural Network Hybridisation

There are three levels at which GA search procedures can be introduced to ANNs, namely, connection weights and biases, architectures and training algorithms. In this work GA has been used to optimise the connection weights and biases of the neural network.

6.2.1 Optimising ANN Weights Using GA

The GA training approach is divided into two major steps. The first one is to decide the representation scheme of connection weights, e.g., binary strings. The second one is the evolution itself driven by GA. Different representation schemes and GAs can lead to quite different training performance in terms of training time and accuracy. A typical cycle of the evolution of connection weights with GA is shown in figure 6.1.

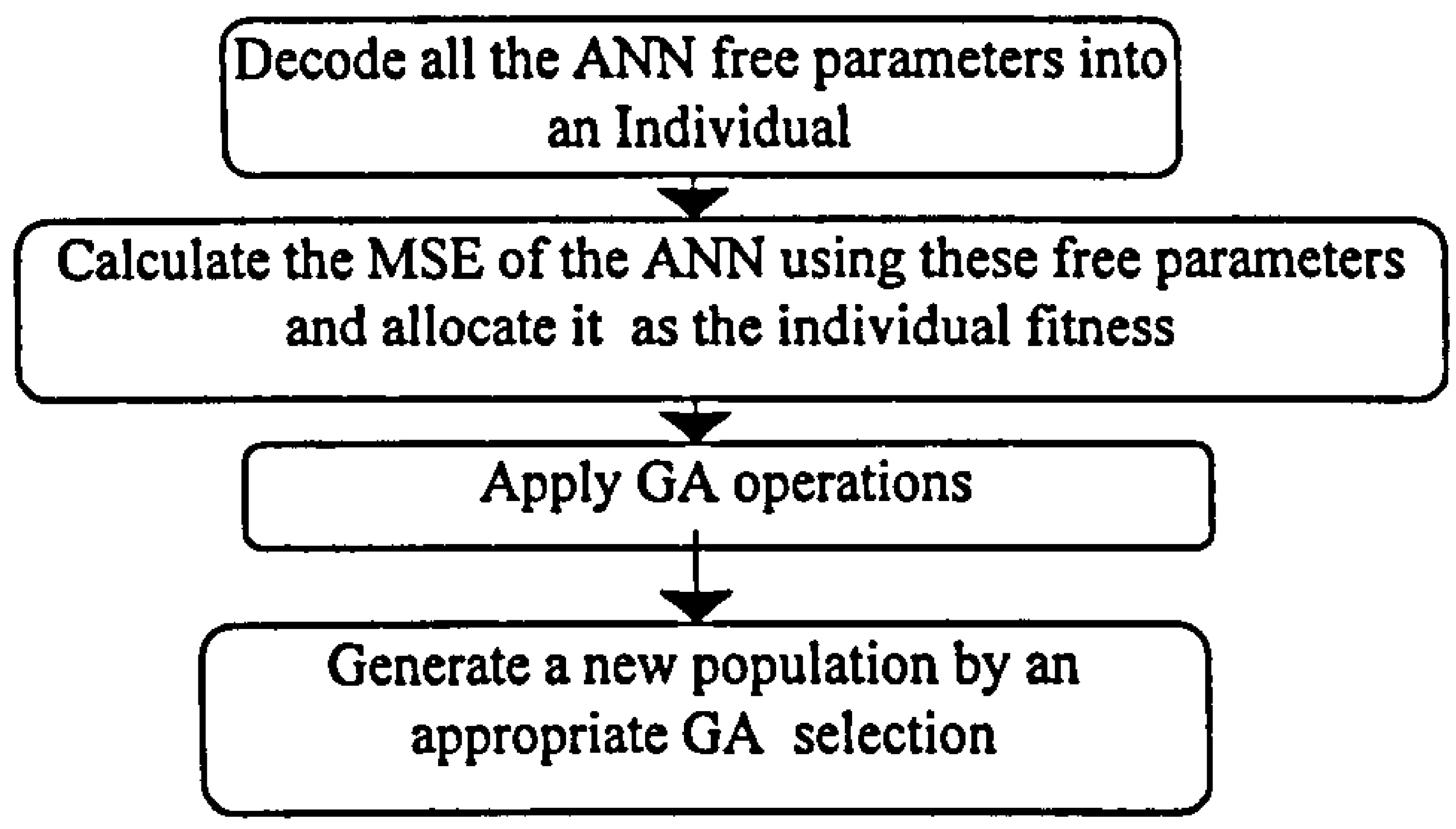

Figure 6.1: Typical cycle of evolution of connection weights with GA

6.2.2 Representation of Connection Weights

When using GA the most convenient representation is binary, since GA uses binary representation (chromosomes) of the problem parameters and binary operators for combination. The range of each free parameter depends on the problem complexity and the required resolution of the system parameters.

A key issue here is to decide how much information about architecture should be encoded into a representation. This includes number of layers and number of neurons in each layer. As the architecture parameters decoded in GA individuals are increased, the computational cost increases. There is a trade off between these two factors as the combination differs for different classes of problems.

6.2.3 GA and ANN Hybridisation

Interaction between developed ANN and GA components are presented in order to explore possible benefits arising from these combinations, instead of using them individually. For this work, software developed by the research team at City University was used. An Object Oriented Technique gives us the ability to combine the existing developed objects and create new components. In order to perform this task, a thorough analysis on both objects should be done including; understanding the principles of the hybrid systems, identifying objects which will remain important in the life of the hybrid system and finally identifying the relationships between the different objects and the ways in which the objects interact.

After analysis of the system, which includes identifying the object interactions, adaptation of classes in the new environment is performed. This task also includes composing ANN free parameters including weights, biases and decoding them into chromosomes. The number of ANN free parameters is calculated as shown in the equation below to form the chromosomes.

$$n_{free} = \left[(n_{in} \times n_{hid}) + (n_{hid} \times n_{out}) + n_{hid} + n_{out}\right] \qquad (6.1)$$

Where:

n_{in} = Number of Nodes in Input Layer, n_{hid} = Number of Nodes in Hidden Layer,

n_{out} = Number of Nodes in Output Layer, n_{byte} = Number of Bytes each Parameter,

n_{free} = Number of Free Parameters, s_{free} = Size of Free Parameters in Bytes

Other parameters such as architecture and training algorithms can be added to the chromosomes as an extension. The interaction between ANN and GA objects is performed by message passing. Both ANN and GA instances are created at the beginning of the optimisation procedure and last until the end. The GA object makes calls to ANN object and passes messages to the fitness function. The optimisation is processed to find the near optimum global solution for each applied problem. Experience shows that

GA_ANNs are highly application dependent, the approach is tested on a parabolic function approximation and on electric load forecasting.

6.3 GA-ANN Application

6.3.1 Parabolic Function Approximation

Parabolic function parameters (x and y values) were obtained using "MatLab" to create training and testing data files. Table 6.1 shows the following system parameters found to be the best for this particular problem.

Table 6.1 Parameters for Parabolic Function Approximation

Parameter	Value
Population size	200
ANN free parameters	16
Bits for each parameter	10
Mutation rate	0.1
Crossover rate	0.9
Number of inputs to the neural network	1
Number of nodes in hidden layer	5
Number of outputs	1

A Similar network was constructed using BP_ANN for comparison with GA_ANN. Figure 6.2 shows comparison of BP and GA training schemes' resulting error functions in the first 100 generations during the training.

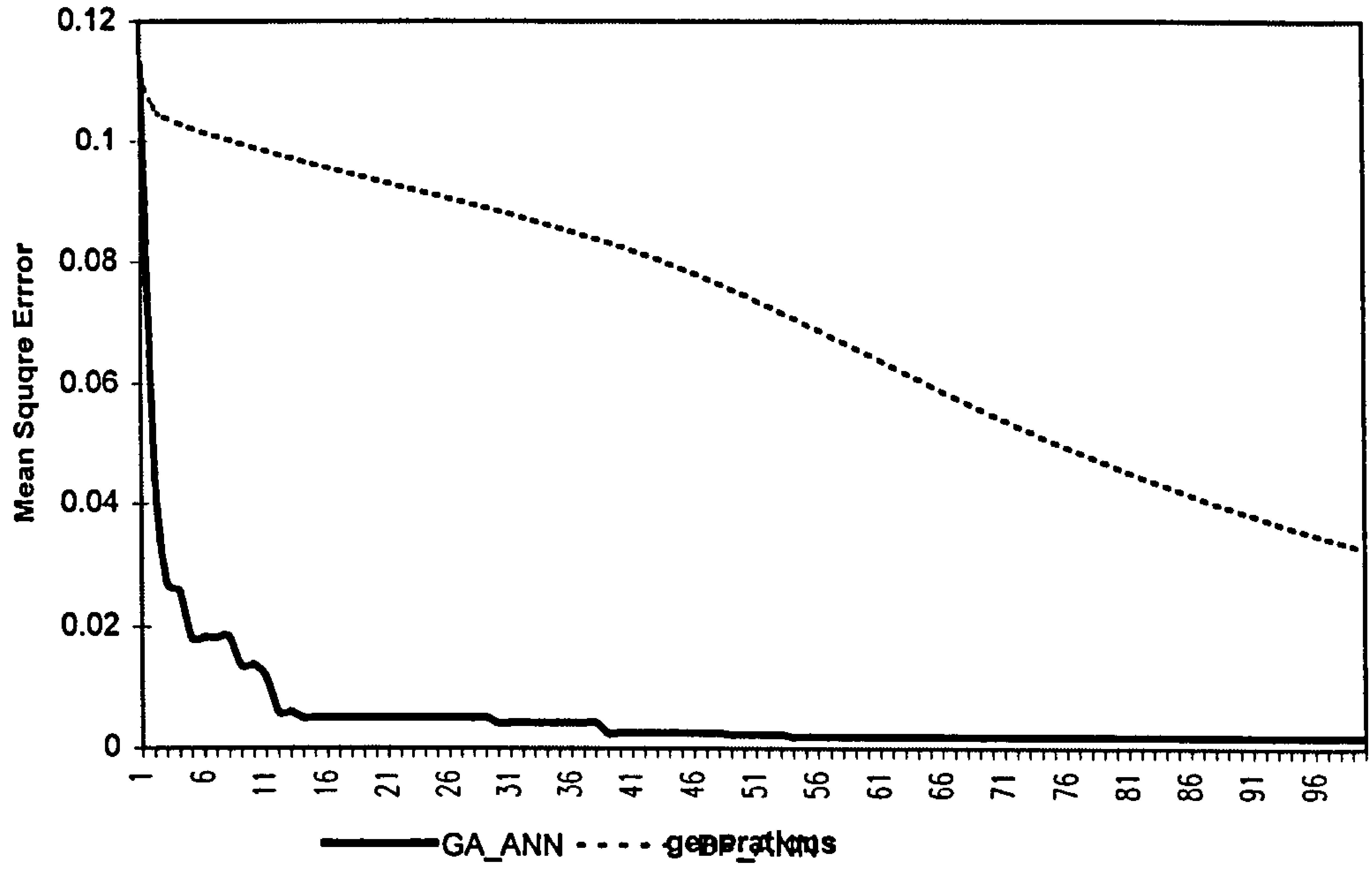

Figure 6.2: Comparison between GA and BP during training for 100 Generations

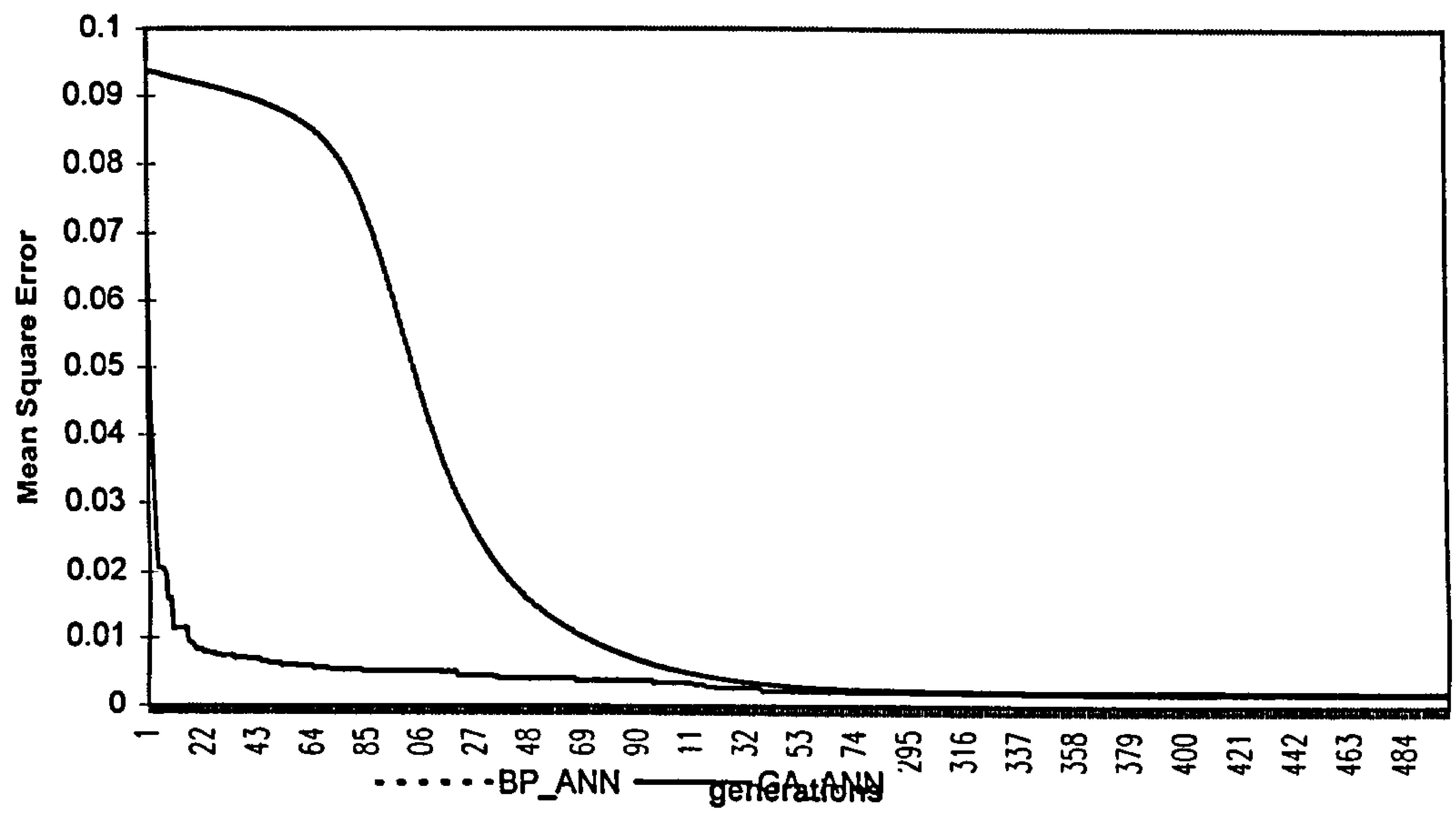

Figure 6.3: Comparison between GA and BP during training for 500 Generations

GA_ANN and BP_ANN were further trained to reduce the RMS error in order to obtain better results. Figure 6.3 shows comparison between BP and GA training scheme's resulting error functions in the first 500 generations during the training.

As shown in Figure 6.2, the GA_ANN system converges much faster than BP_ANN, meaning that it finds a near global optimum, there is no significant difference in computational time. The GA training scheme shows improvement over the BP in the first 100 generations and presents a very good convergence. However, when GA generations increase, the convergence speed reduces rapidly.

Figures 6.4 and 6.5 show the outputs of GA_ANN and BP_ANN for unseen data of a parabolic function with 100 and 500 generations of trained GA_ANN and BP_ANN respectively.

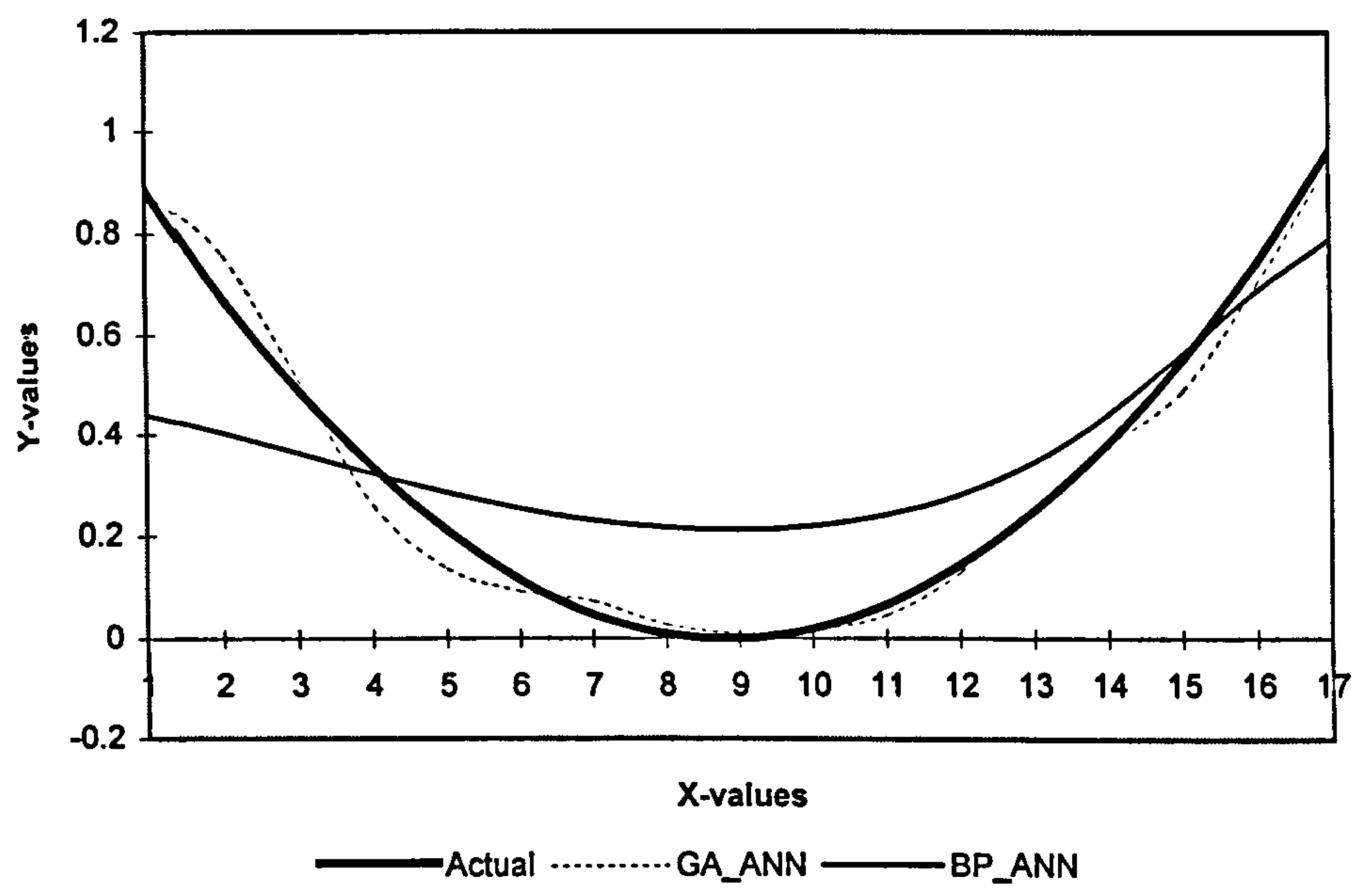

Figure 6.4. Comparison between Actual, GA_ANN and BP_ANN outputs for unseen data with 100 generations of trained GA_ANN and BP_ANN

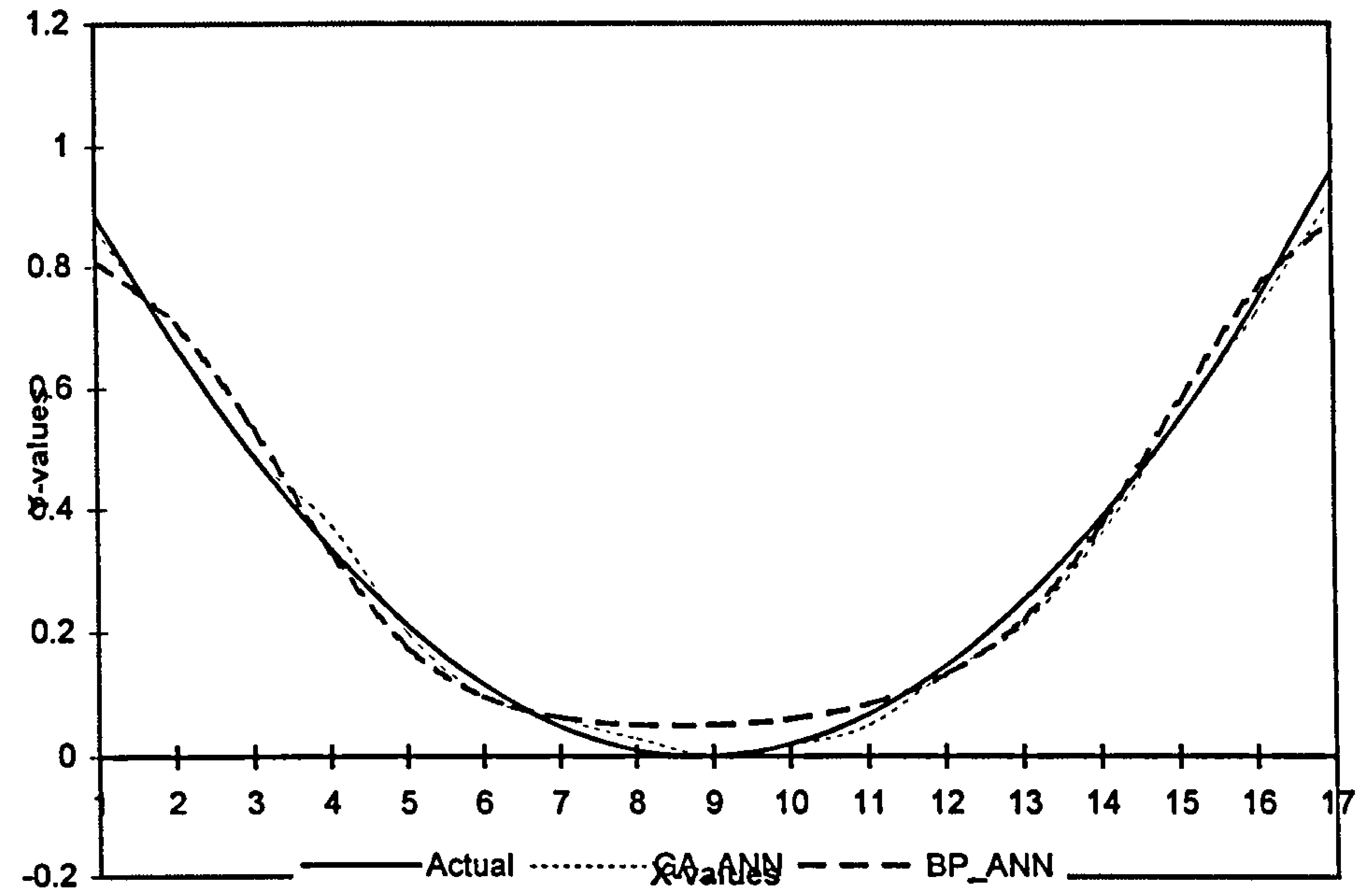

Figure 6.5: Comparison between Actual, GA_ANN and BP_ANN outputs for unseen data with 500 generations of trained GA_ANN and BP_ANN

6.3.2 Short-term Load Forecasting

There are 58 inputs to the developed model. The features that are taken into account as input factors in the load forecast system are as follows: two 24-hour load records of day *i*-1 and *i*-2 (the forecast day is day *i*). Six more inputs are the maximum and minimum temperatures of day *i*, *i*-1 and *i*-2. Three other inputs are the binary codes to show seven days of the week. One binary code is dedicated to the holidays or any yearly special occasions that may affect the forecast. In summary, the designed ANN is of the MLP type and is used to learn about the relationship between the 58 inputs and 24 outputs. The data for this study was obtained from an Italian Power company and was available in the public domain (Internet).

The inputs are:

Hourly loads for two days prior to the forecast day	24
Hourly loads for the day prior to the forecast day	24
Max. and Min. temps for two days prior to the forecast day	4
Max. and Min. temps for the forecast day	2
Day of the week	3 bits
Holiday	1 bit

The outputs are:

Load forecast for all 24 hours of the day	24

The above values are normalised as indicated by Equation (6.2).

$$\text{Normalised} = \frac{\text{Actual Value} - \text{Min.}}{\text{Max.} - \text{Min.}} \tag{6.2}$$

where Max. and Min. are the maximum and minimum of the attribute, respectively.

The mean square error (MSE) is used to measure the accuracy of the model. The sigmoid activation function is adopted.

The following system parameters are found to be the best for this problem and are shown along with their optimum values in Table 6.2.

Table 6.2 ANN Input parameters

System Parameters	Value
population size	150
ANN free parameters	1269
bits for each parameter	16
mutation rate	0.1
crossover rate	0.9
number of inputs to the neural network	58
number of nodes in the hidden layer	15
number of outputs	24

The Back-propagation was also used to train another ANN which is then used as a reference to make a comparison between two algorithms. The training Mean Square Error (MSE) of both BP_ANN and GA_ANN for the first 100 generations is shown in figure 6. 6. The BP_ANN converges much faster than GA_ANN. It could mean that it finds a better optimum in less number of iterations. Figure 6.7 shows the comparison between actual results and GA_ANN outputs for unseen data.

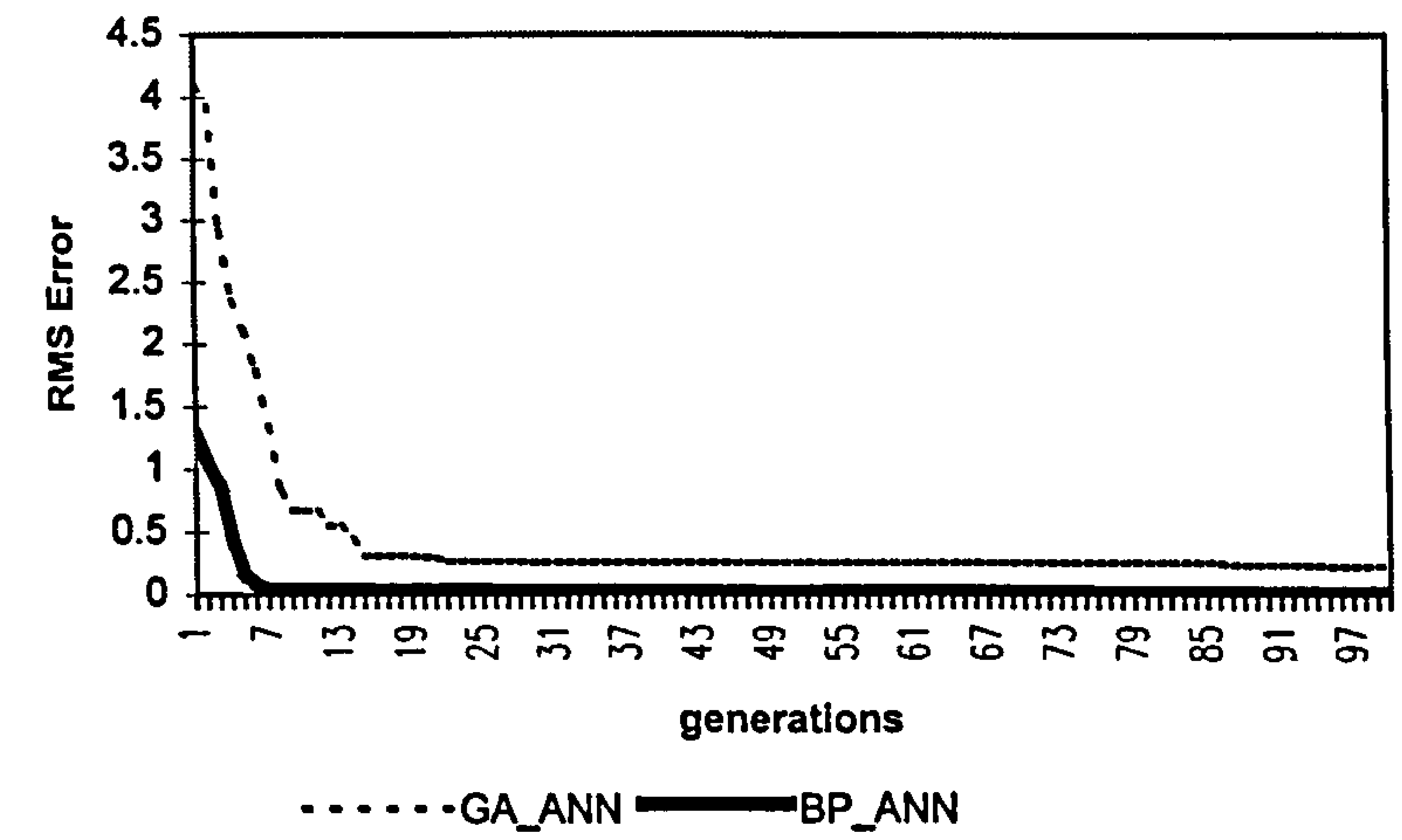

Figure 6.6: Comparison between GA_ANN and BP_ANN during training for 100 generations

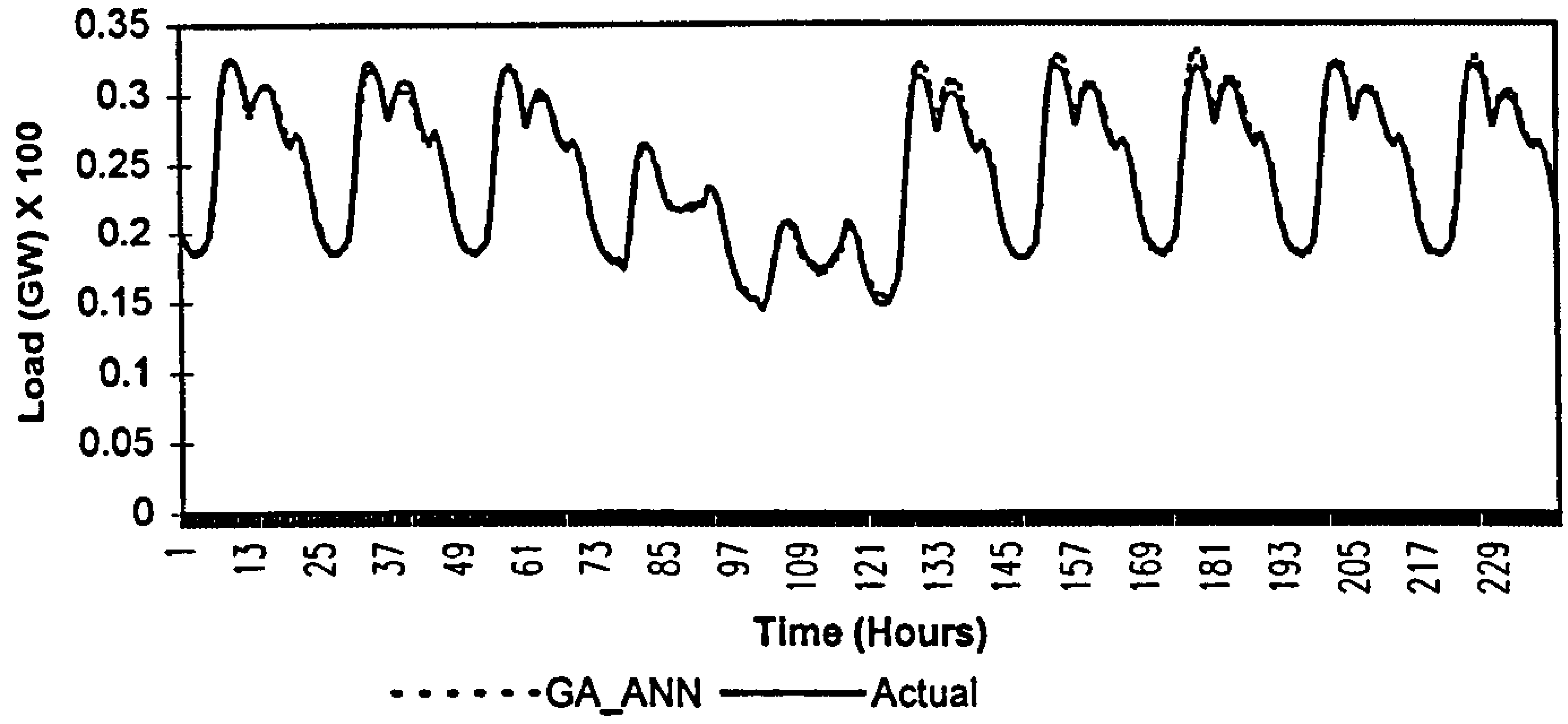

Figure 6.7: Comparison between Actual and GA_ANN outputs for unseen data

6.4 Conclusions

The main question is whether the GA_ANN is more efficient than the conjugate gradient (e.g. BP) methods. In general GA_ANN gives better solutions for the problems with a small number of parameters. But for systems with a large number of problem parameters it becomes impractical as it increases the computational time and the computational cost. If powerful computer facilities are available, then GA_ANN are generally the preferred method. Parallel GA_ANN is one of the solutions to reduce training time. The Object Oriented methodology is a very useful framework in the development of GA_ANN as it reduces the development time. The Object Oriented models of all algorithms give this flexibility to upgrade and maintain the software constantly and form different configurations. Artificial Neural Network and a Genetic Algorithm have been used to design a neural network for short-term load forecasting. The forecasting model has been used to produce a forecast of the load in hour intervals of the forecasted day, using data provided by an Italian power company. The BP_ANN will give a local minimum but a ANN_GA will give the best solution, because of the local minimum problem the BP_ANN is not suitable for load forecasting. The results obtained are promising. In this particular case, the comparison between the results from the GA_ANN and BP_NN shows that the GA_ANN does not provide a faster solution than the BP_NN. This could be due to the fact that the initial randomly selected starting point is a poor one, poor starting will take a long time to converge. As the size of the problem is very large the amount of memory and computation time required is large too. This points to the direction of parallel processing techniques being integrated with evolutionary computing to solve complex practical problems.

Chapter VII

APPLICATION OF EVOLUTIONARY PROGRAMMING TO FAULT SECTION ESTIMATION

7.1 Introduction

To enhance service reliability and to reduce power outage time, rapid restoration of a power system is required. As a first step of restoration, the fault section should be quickly estimated as accurately as possible. The Fault Section Estimation (FSE) identifies fault components in a power system by using information from the operation of protection relays and circuit breakers. However this task is difficult especially for cases where the relay or circuit breaker fails to operate and for multiple faults.

Several papers have reported surveys on Evolutionary Algorithms (EA) applications in power systems [83,84]. Few methods have so far been employed to solve the FSE problem. This includes:- Expert Systems [85] and other Computational Intelligence Techniques (CIT), such as, Artificial Neural Networks [86,87] and Genetic Algorithms [88]. Among these methods, expert systems are based on the production of a great number of rules describing the complex protection system behaviour. This will result in the following problems:

1. The maintenance of such a complex rule based knowledge base is very difficult, costly and time consuming.
2. The response time for complex systems is long.

The application of ANN to FSE is an active research area. However, the correctness of the estimation result cannot be proved theoretically, as a result the system is questionable. As the FSE objective function is usually a high order polynomial, the GA optimisation method has been employed to deal with such a problem [88]. Evolutionary Programming is an optimisation algorithm based on the mechanics of natural selections, mutation, competition and evolution. The process of evolution inevitably leads to the optimisation of "behaviour" within the context of a given criterion. EP does not require crossover operation and it has a shorter run time when compared with GA [89].

This is the first time evolutionary programming has been used for FSE. This chapter is presented in the following sections. After formulating the problem in section 7.2, the derivation of the fitness function and EP parameter is given in section 7.3. The EP Algorithm is described in section 7.4 and a case study for a modelled power system is introduced and examined by the developed algorithm in section 7.5. The conclusion is given in section 7.6.

A C++ code has been developed to implement the proposed EP and GA. A Visual C++ compiler was used on a Windows NT platform running on a Pentium 166 machine.

7.2 Problem Formulation

To simplify the explanation of the proposed approach, a model is developed. It is assumed that the modelled power system consists of the following components and parameters:

- Circuit Breaker (CB),
- Relay (R),

- Busbar (BB),
- Transformer (T),
- Transmission Line (L), and
- Section i (A_i)
- There are three types of relays :
  - Main Protection Relay(MPR),
  - Primary Backup Protection Relay (PBPR), and
  - Secondary Backup Protection Relay (SBPR).

The operation and description of the protection relays are as follows.

1. Each **Busbar** has one MPR. It is used to initiate the circuit breakers to disconnect the fault in the busbar.
2. Each **Transformer** has three relays, namely, MPR, PBPR and SBPR. **MPR** is used to initiate the two Circuit Breakers at its ends. **PBPR** is to initiate the Circuit Breakers when a fault is on one of its neighbouring elements while the main protection relay fails to operate. The purpose of **SBPR** is to protect the transformer in case of a fault occurring on one of its neighbouring elements and the main protection relay of the faulted element fails to operate.
3. Each **Transmission Line** has two sets of MPR, PBPR and SBPR; one for the sending end and one for the receiving end. **MPR** of each end is to actuate the Circuit Breaker of that end when there is fault on the line. **PBPR** in each end is used to protect the line in case a fault has occurred while the main relay fails to operate. **SBPR** is used to

protect the transmission line in case a fault occurs on one of its neighbouring elements but the main protection relay fails to operate.

7.3 Mathematical Model and Fitness Function

In order to use the EP technique, the problem has to be modelled mathematically. The EP technique is based on the assumption that a fitness landscape can be characterised in terms of variables and there is one or a set of optimum solutions. A mathematical model has been introduced in [87] and is used to formulate a 0-1 integer programming problem which is then solved by GA. Although GA and EP have similar data dictionaries, they have a different data structure. EP has more flexibility for selection of data type. It operates directly on individual type e.g. floating-point. A form of the same mathematical model used in [87] is adapted in this chapter. This modified version can be operated in conjunction with EP characteristics in the forms of a [min, max] floating point programming problem.

The fitness function is one of the main elements of an EP algorithm. Fitness function evaluates each individual and returns a value indicating how fit that individual is, to be considered as a solution to the problem.

Each component of the system corresponds to a mathematical term which appropriately affects the final value of the fitness function. The status of busbars and relays are presented by two values, 0 for non operational and 1 for operational conditions. Each section of the power system model is considered as an individual for EP. This individual varies over a predefined range - $A_{\min} < A_i < A_{\max}$. In this chapter $A_{\max}$ is 1 and $A_{\min}$ is 0. The combination of 0-1 integers from relays and busbars status and floating point value of individuals created by EP, will result in an appropriate value to show the fitness of that particular individual. The final output of the fitness function will then be deducted from a large positive constant number in order to secure a positive fitness.

By considering the above points, the corresponding mathematical terms for each Relay are directly related to the protection system, explained in section 7.2 and are listed as follows:

- MPR of any busbar, transformer, sending and receiving ends of transmission line. The equations ensure a positive fitness function:

$$\left(1-2\left(A_i\_MAIN\right)\right)A_i \tag{7.1}$$

$$\left(1-2\left(T_i\_MAIN\right)\right)T_i \tag{7.2}$$

$$\left(1-2\left(L_iS\_MAIN\right)\right)L_i \tag{7.3}$$

$$\left(1-2\left(L_iR\_MAIN\right)\right)L_i \tag{7.4}$$

Where *MAIN* shows the main protection relay of each component and A_i, T_i, and L_i are the sections.

- PBPR of any transformers, sending and receiving ends of each transmission line:

$$\left(1-2\left(T_i\_PRIM\right)\right)T_i\left(1-T_i\_MAIN\right) \tag{7.5}$$

$$\left(1-2\left(L_iS\_PRIM\right)\right)L_i\left(1-L_iS\_MAIN\right) \tag{7.6}$$

$$\left(1-2\left(L_iR\_PRIM\right)\right)L_i\left(1-L_iR\_MAIN\right) \tag{7.7}$$

Where *PRIM* and *MAIN* show the primary back-up and main protection relays respectively for each component. T_i, and L_i are the sections.

- SBPR of any transformer:

$$\left(1\text{-}2\left(T_i\_SEC\right)\right)\times\left\{1-\left[1-A_j\left(1-A_j\_MAIN\right)\right]\left[1-A_k\left(1-A_k\_MAIN\right)\right]\right\}\times\left(CB_l|CB_n\right) \quad (7.8)$$

Where T_i_SEC is the SBPR of transformer T_i, A_j is one of the sections protected by this relay, A_j_MAIN is the Main relay of this section, A_k is the other section which is protected by this relay and A_k_MAIN is its Main Relay. CB_l and CB_n are neighbouring circuit breakers of the transformer.

- SBPR of any transmission line:

For sending end:

$$\left(1-2\left(L_iS\_SEC\right)\right)\left\{1-\left[1-A_j\left(CB_k\right)\right]\right\} \quad (7.9)$$

For receiving end:

$$\left(1-2\left(L_iR\_SEC\right)\right)\left\{1-\left[1-A_j\left(CB_k\right)\right]\right\} \quad (7.10)$$

Where L_iS_SEC and L_iR_SEC are the secondary sending and receiving end relays respectively. CB_k is the corresponding circuit breaker of the corresponding section A_j. Each configuration of circuit breakers and relays produces an input Pattern. A list of possible fault sections forms the system output.

7.4 EVOLUTIONARY PROGRAMMING

EP is a computational intelligence method in which an optimisation algorithm is the main engine for the process of three steps, namely, natural selection, mutation and competition. According to the problem, each step could be modified and configured in order to achieve the optimum result. Each possible solution to the problem is called an individual. The mathematical form of the i individual is

$$P_i = \left[A_k^i\right], k = 1,2,...m \qquad (7.11)$$

where P_i is 'where' the fault is A_k is the section that the fault is in, m is the maximum number of parameters in any possible solution and $A_{\min} < A_k^i < A_{\max}$. In order to use EP, the mathematical model should be capable of dealing with the data type and structure of individuals.

7.4.1 Initialisation

The initial population consists of individuals (sections) and is created randomly. The fitness score f_i of each p_i is obtained by a fitness function. The fitness function is the summation of the corresponding terms of each protection relay and circuit breakers. The rest of the EP procedures are same as before.

7.5 Genetic Algorithm

Genetic Algorithms are the most popular and widely used of all the evolutionary algorithms. They have been widely applied to solve complex non-linear optimisation problems in a number of engineering disciplines. They operate on population of strings (chromosomes) that encode the parameter set of the problem to be solved over some finite alphabet. In the chosen case study, there are 10 parameters. Each parameter is encoded into an 8 bit binary string (gene) producing an 80 (8 x 10) bit chromosome. As the range of each parameter is from 0 to 1, this gives us the step size (precision) of 0.0039, which is quite acceptable for this case. Each encoding represents an individual in the GA population. The population is initialised to random individuals (random chromosomes) at the start of the GA run. The GA searches the space of possible chromosomes (hamming space) for better individuals. The search is guided by "fitness" values returned by the modelled protection strategy explained earlier. The same fitness function as used in EP is adopted. This gives a measure of how well each individual is in terms of solving the problem and hence determining its chance of appearing in future generations. Two-point crossover has been used and the crossover probability and mutation probability are 0.85 and 0.1 respectively. The Elitist generation method has been used to transfer the best individual through the generations and therefore guarantee the convergence.

7.6 Evolutionary Programming Vs. Genetic Algorithms

Considering both algorithms, it can be concluded that the EP run needs less computational time than the GA run. This has also been proven by the tests using the developed C++ code for each algorithm. The difference in computational speed can be related to the characteristic of each algorithm. GA individuals are represented in a binary form, hence crossover and mutation are binary operators. Execution of these operators takes a longer time in comparison to the simple EP mutation operator, especially when the number of system free parameters is high and chromosomes are long and/or population size is large. The following tests can demonstrate the efficiency of each algorithm for the specific problem. In the following section the results with both EP and GA are presented. The presented graphs that show the variation of the maximum fitness during the generations can be used to compare the behaviour of EP and GA.

7.7 Case Study

Figure 7.1 shows a power network which is used to demonstrate the capability of the EP-based algorithm. The network consists of 10 sections, corresponding to 28 Relays and 14 Circuit Breakers.

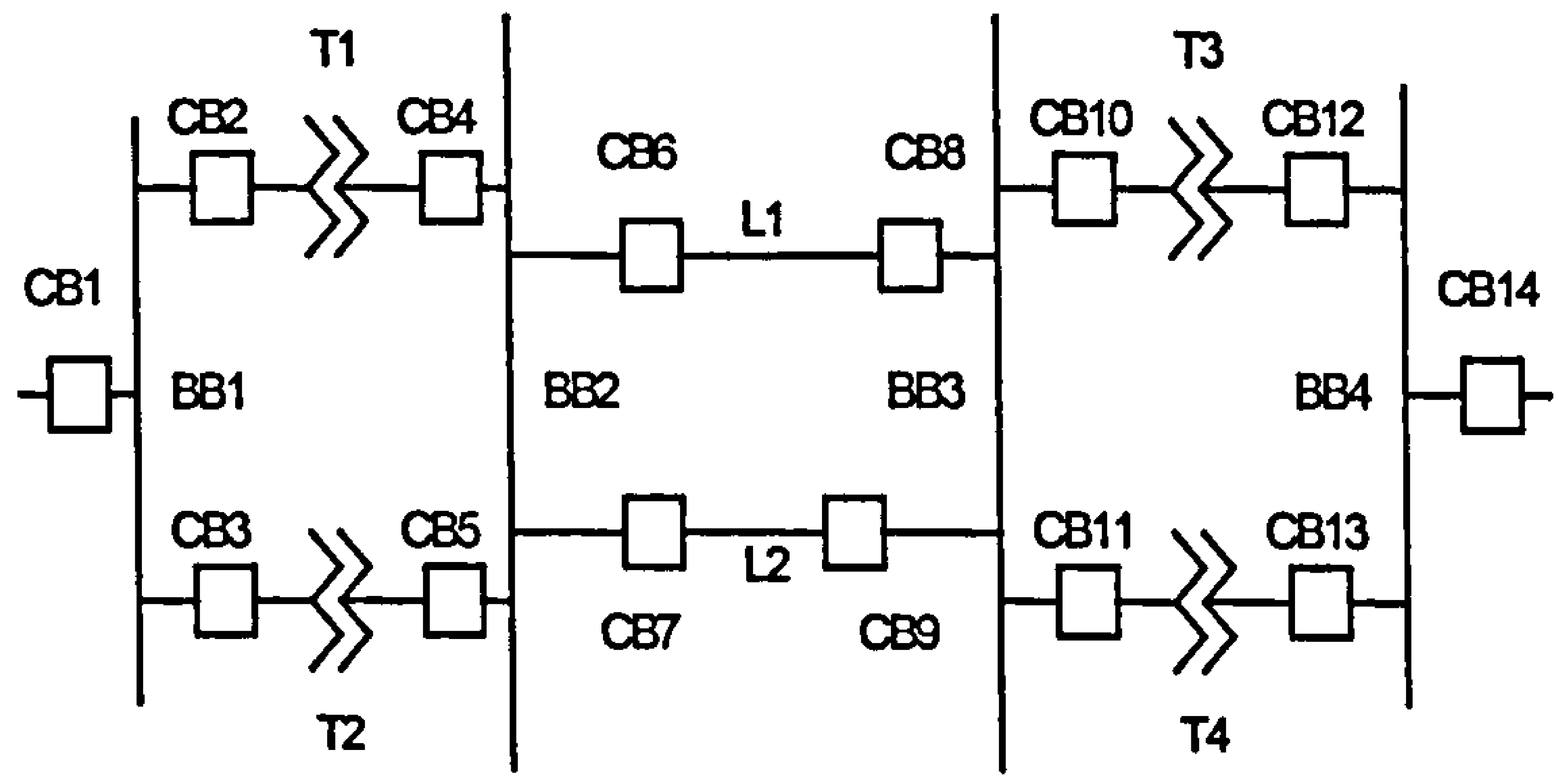

Figure 7.1: Power network

The sections, protection relays and their assignments are as follows:

A1-A10: BB1, BB2, BB3, BB4, T1, T2, T3, T4, L1, L2.

R1-R12: BB1_MAIN, BB2_MAIN, BB3_MAIN, BB4_MAIN, T1_MAIN, T2_MAIN, T3_MAIN, T4_MAIN, L1R_MAIN, L1S_MAIN, L2R_MAIN, L2S_MAIN.

R13-R28: T1_PRIM, T2_PRIM, T3_PRIM, T4_PRIM, T1_SEC, T2_SEC, T3_SEC, T4_SEC, L1R_PRIM, L1S_PRIM, L2R_PRIM, L2S_PRIM, L1R_SEC, L1S_SEC, L2R_SEC, L2S_SEC.

Three different cases have been considered and the fitness has been calculated for each of them. The conditions and results are shown in Tables 7.1 to 7.3 and figures 7.2 to 7.4.

For case 1, the operated relays and circuit breakers are given in Table 7.1. The number of generations, population size and estimated fault sections are also presented in the same table. Figure 7.2 shows the average, maximum, minimum and difference between maximum & minimum fitness.

Table 7.1. The status of relays & circuit breakers and results [from reference [82], reproduced by permission of IEE, UK]

Gen. No.	**Size of Pop.**	**Actuated Relays**	**Actuated Circuit Breakers**	**Results: Fault Section(s) EP**	**Results: Fault Section(s) GA**
200	10 & 50	L1R_SEC, L2R_SEC, T4_SEC	CB8,CB10,CB11	BB3	BB3

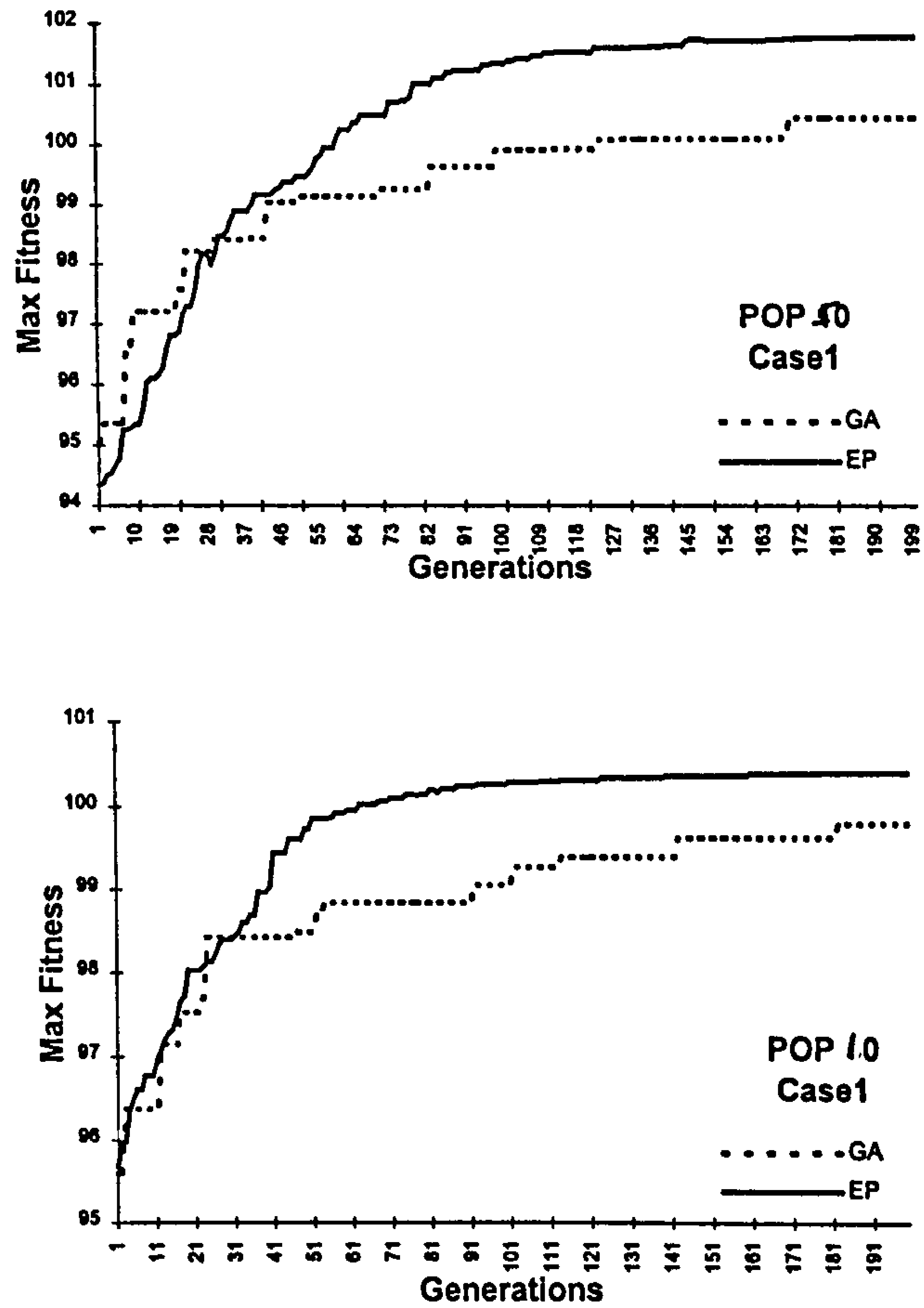

Figure 7.2: Maximum fitness against number of generations, case 1

In this case, a possible explanation could be as follows:

BB3_MAIN, L1S_SEC & L2S_SEC fail-to-trip. CB10 is actuated, so T3_SEC may operate but the signal has not been received by the control centre. CB9 may be tripped by L2R_SEC. CB9 operates but no signal has been received by the control centre too.

For case 2, the operated relays and circuit breakers are given in Table 7.2. The number of generation, population size and estimated fault sections are also presented in the same table. Figure 7.3 shows the average, maximum, minimum and difference between maximum & minimum fitness.

Table 7.2: The status of relays & circuit breakers and results

Gen. No.	Size of Pop.	Actuated Relays	Actuated Circuit Breakers	Results: Fault Sections EP	Results: Fault Sections GA
200	10 & 50	T1_SEC, T2_SEC, L1S_MAIN,L1R_ MAIN, L1S_PRIM	CB4, CB5 CB8	BB2, L1	BB2, L1

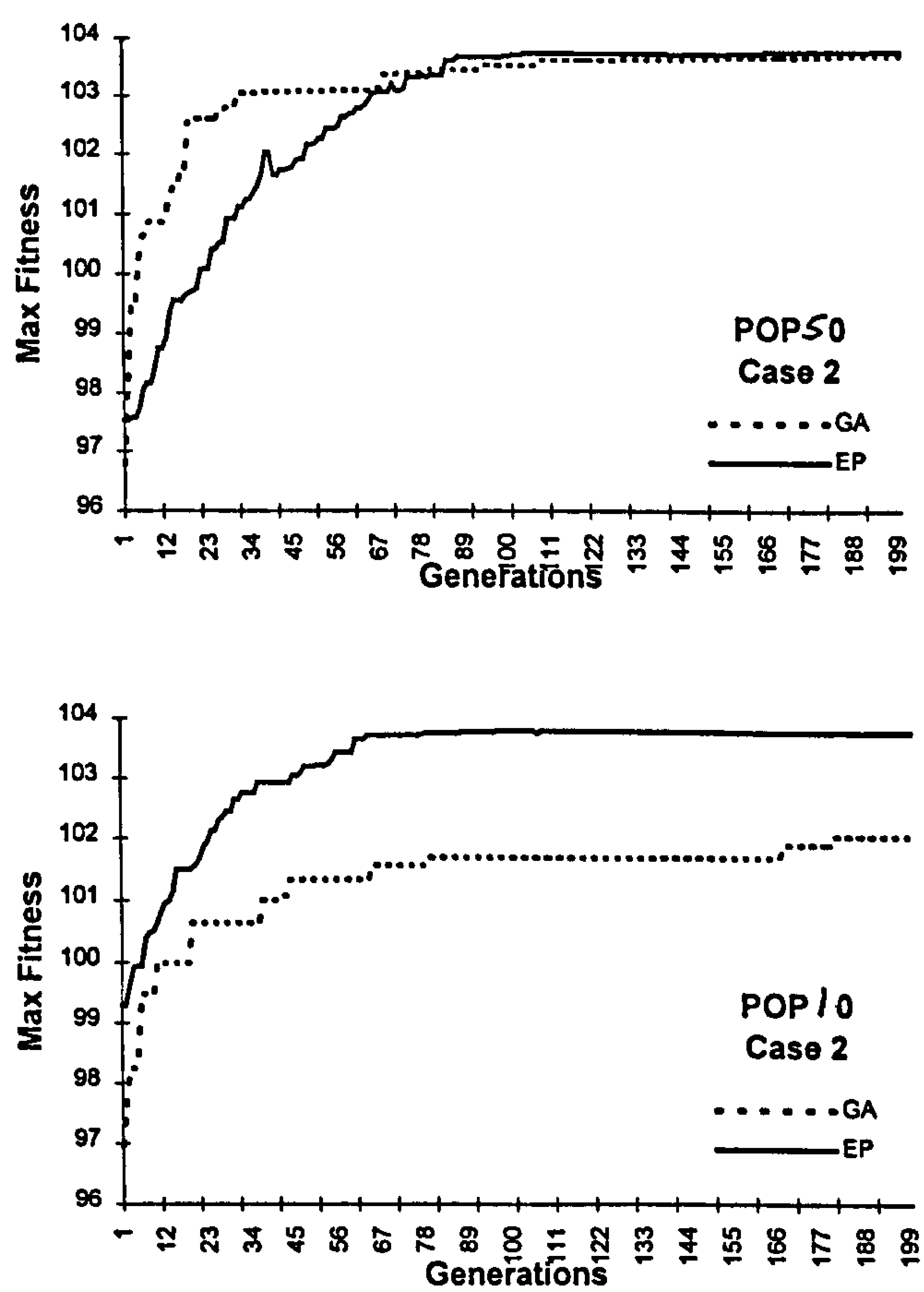

Figure 7.3: Maximum fitness against number of generations, case 2

For case 3, the operated relays and circuit breakers are given in Table 7.3. The number of generation, population size and estimated fault sections are also presented in the same table. Figure 7.4 shows the average, maximum, minimum and difference between maximum & minimum fitness. The results show that there is a higher chance for the fault to occur in section T4.

Table 7.3: The status of relays & circuit breakers and results

Gen. No.	Size of Pop.	Actuated Relays	Actuated Circuit Breakers	Results: Fault Sections EP	Results: Fault Sections GA
200	10 & 50	T3_SEC, T4_PRIM, BB4_MAIN	CB13, CB14, CB10	BB4, T4	BB4, T4

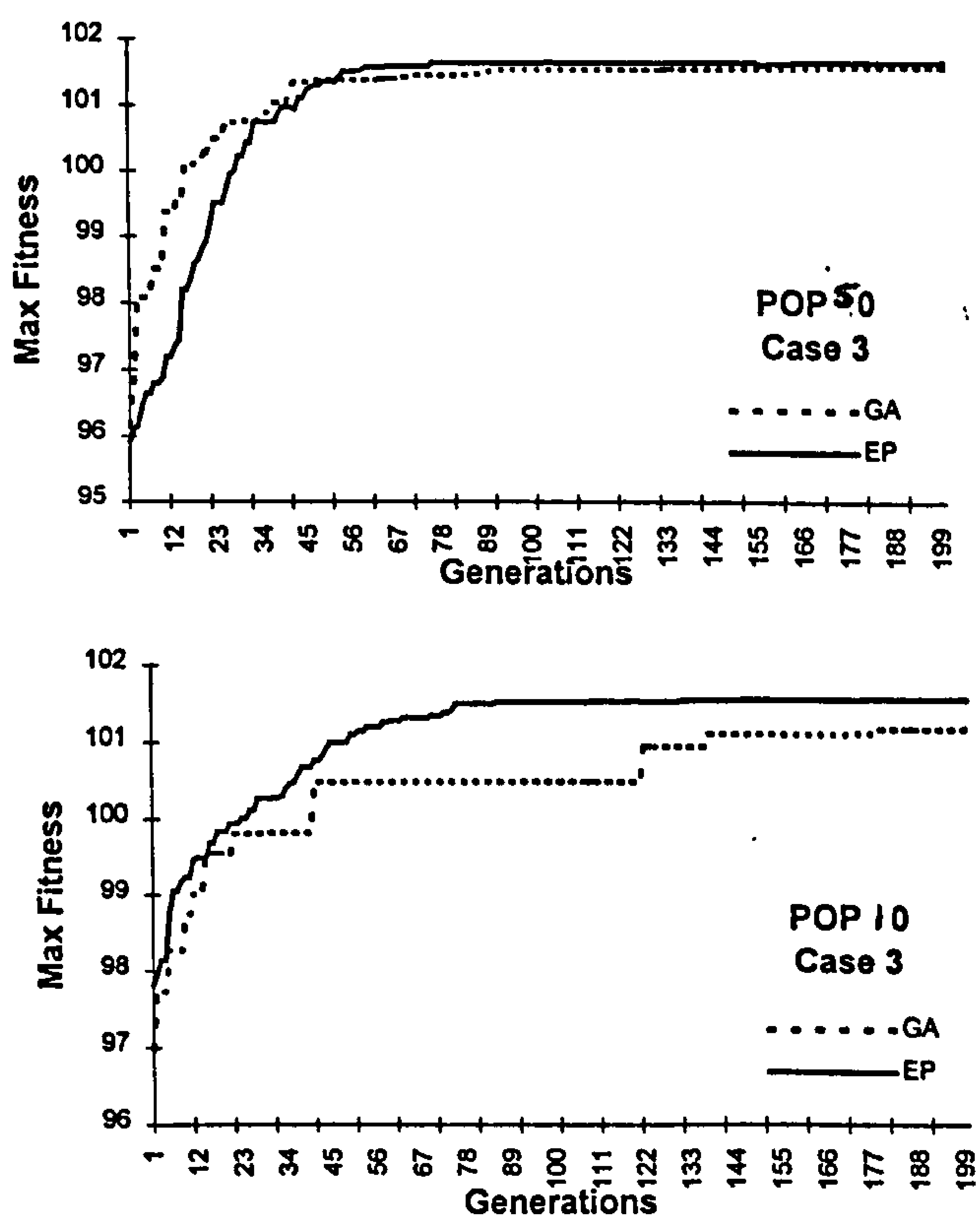

Figure 7.4: Maximum fitness against number of generations, case 3

This computer simulation on average will give an optimum result in about 60 to 100 generations. The population size is as small as 10, which speeds up the processing time. All the simulation results are performed on a Pentium 166 MHz PC. The CPU time for each case is about 2.5 seconds.

7.8 Conclusions

An EP approach has been developed for solving the FSE problem including malfunctions of protection relays and/or circuit breakers and multiple fault cases. A comparison is made with the GA approach at the same time. Two different population sizes are tested for each case. In general, EP showed faster computational speed than GA with an average factor of 13 times more. The final results were almost the same, the results may not be the same if this technique was applied to other problems, therefore comparisons should be done in the same problem domain. The convergence speed (the required number of generations to get an optimum result) is a very important factor in real time applications. Test results show that EP is better than GA. However, as both EP and GA are evolutionary algorithms, their efficiencies are largely dependent on the complexity of the problem which might differ case to case. EP is also ideal for parallel processing computer systems or hardware. Therefore, with this kind of it is possible to solve FSE problem faster and with high efficiency.

Chapter VIII

The Use of Artificial Neural Networks to Classify Faults from Digital Fault Records

8.1 Introduction

Digital fault records are a very useful source of information to the protection engineer to assist with the investigation of a suspected unwanted operation or failure to operate of a protection scheme. After a wide spread power system disturbance due to a storm for example, a large number of fault records can be produced. A method of automatically classifying fault records would be very helpful in reducing the amount of time spent in manual analysis, thus assisting the engineer to focus on records that need in depth analysis. Fault classification using rule base methods have already been developed.

A noticeable advantage of an ANN over rule based systems is that they are capable of dealing with input vectors of data that are partially incomplete or incorrect. A trained ANN is capable of doing this because what it learns about one pattern generalises to other similar patterns. Conflicting information does not paralyse the ANN , it will still make its best judgement based on the information available. On the other hand, the rule based system will typically fail if not presented with complete and accurate input data. An ANN could therefore compliment a rule based system for classifying digital fault records, in offering the protection engineer a possible solution when the rule based system fails to recognise a particular fault.

8.2 Rule Based Automatic Analysis of Fault Records

8.2.1 Automatic Analysis of Protection Response

Fault recorders deployed throughout a power transmission system are usually triggered to provide records when circuit breakers trip and when disturbances cause protection relays to respond [90]. Many fault records can be generated by one system fault or incident and particularly following multiple faults caused by adverse weather conditions, the protection engineers have to deal with a large numbers of fault records. For each record the engineer needs to determine if the protection responded correctly, and also to enter the data from the record into a fault analysis and defect system. This process is very time consuming. In order to reduce this time and to assist the engineer in focussing on records which need in-depth analysis, The National Grid Company has developed rule based software which automates this analysis.

The software can analyse a batch of fault records extracted from a number of different types and manufactures of fault recorders. The output of the automatic analysis consists of various fields of information which can be selected as required.

(i) Disturbance classification

A classification is given to the disturbance which caused the recorder to trigger. E.g. Type of fault and if internal or external.

(ii) Analysis of protection response

A logic and timing analysis is carried out and if incorrect or out of limits this is indicated.

(iii) Fault statistics

Automatic entry of fault record information into NGC's system of protection performance statistics.

8.3 An Artificial Neural Network Approach to the Analysis of Digital Fault Records

8.3.1 Self Organising Map (SOM)

SOM networks create a two-dimensional feature map (represented in the network by the "Kohonen" layer) of the multi-dimensional input data. In this way, if two input data sample sets are "alike", in terms of the inter-relationships between the various individual data set parameters, then they will be mapped to processing elements that are physically close to each other within the two-dimensional feature map (they are often mapped to the same processing element). The node to which a particular individual data sample set is mapped is termed the "winner", as its output is set to 1, all other nodes in the map having an output of 0.

Thus, the SOM network may be able to subdivide the multi-dimensional input data set into categorical subsets. These subsets are unlabelled, i.e. the network cannot explain the defining parameters for the individual categories. This labelling of the categories must be performed by the user, using prior knowledge of the input data set characteristics. The principles of a SOM network can be more clearly illustrated graphically. Figure 8.1 shows a successfully trained network, which has recognised 4 sub-categories within a set of data containing individual 3 dimensional data vectors (these are the individual train/test examples).

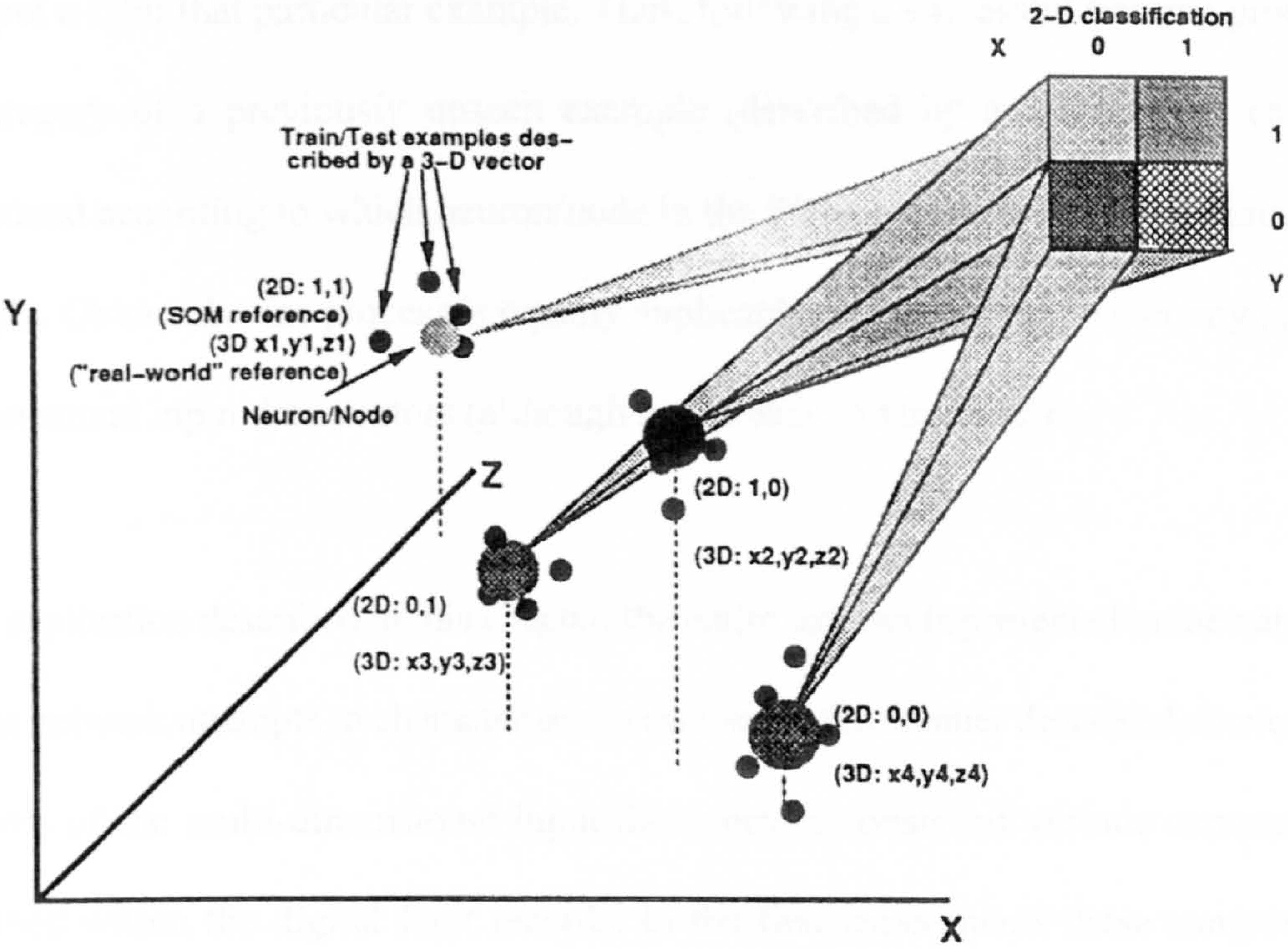

Figure 8.1: SOM after successful training

The individual categories can be seen to fall into the same area within the Euclidean space. The naming of the categories is performed by the user, employing prior knowledge of the categories contained within the entire data set (e.g. in the application discussed in this paper, the categories would be: red-earth internal disturbance, red-earth external disturbance, blue-earth internal disturbance...etc.).

In the example illustrated in Figure 8.1, the ANN would consist of an input layer of 3 nodes (for each element of the 3 dimensional vector) and a Kohonen layer of 4 nodes. Each of the train/test examples, uniquely described by a 3-D vector (e.g. x1,y1,z1), would cause the "closest" neuron/node (i.e. the node with the shortest Euclidean distance between its 3-D co-ordinates and the train/test example's 3-D co-ordinates) in the SOM to output a 1 for that particular example. Thus, following a successful training process, the category of a previously unseen example (described by a 3-D vector) can be determined according to which neuron/node in the SOM outputs a 1 for that particular example. Obviously, the process is equally applicable (and more useful) for any higher n-dimensional input data vectors (although not as easy to visualise!).

In the application described in this chapter, the entire data set is presented to the network and the network attempts to characterise the data set in the manner described above. The elements of the multi-dimensional input data vectors consist of various components contained within the digital fault records. In the first experiments these components consisted of "pre" and "during" disturbance rms amplitude ratios for the currents and voltages. It is proposed that in future, this will be extended to include the frequency components contained within the aforementioned currents and voltages.

In order to test the network, the subsets of data representing the various primary system fault conditions are presented to the trained network individually and the network's response analysed in terms of the distribution of winning nodes. If the network has successfully learned how to classify the data, then the differences between the various sub-classes of input data will be physically reflected in the distribution of the winning nodes throughout the map upon presentation of the data subsets.

8.3.1.1 Experiments to Classify Digital Fault Records

The data for this experiment is a selection of digital fault records down loaded from fault recorders located on The National Grids Transmission network. The train/test data consisted of a total of 33 diagnosed primary system fault records and a total 17 other records relating to minor disturbances, fault recorder tests, Capacitive Voltage Transformer (CVT) disturbances and records which could not be classified. This data was converted to a multi-column ASCII file format to allow ease of data analysis, each column relating to measured analogue values, relay status changes, etc. The main descriptors of primary system fault class are the three phase current and voltage measurements, along with a measure of residual (i.e. neutral) current. It is the evaluation of these seven parameters that proves to be the most useful, when attempting to classify the observed fault or disturbance. The distribution of the available digital fault records with respect to fault class is shown in Table 8.1.

Table 8.1: Diagnosed primary system faults

Fault Type	**Number of Faults**
Red phase to earth	6
Yellow phase to earth	12
Blue phase to earth	2
Yellow phase to red	3
Blue phase to red	2
Blue phase to yellow	2
Three-phase faults	2
Failed circuit closure attempts	4

NB: red = A phase, yellow = B phase and blue = C phase

A typical fault record displayed on the fault recorder manufacturer's replay software is shown in figure 8.2 (In this case CSD Hathaway Systems, Belfast)

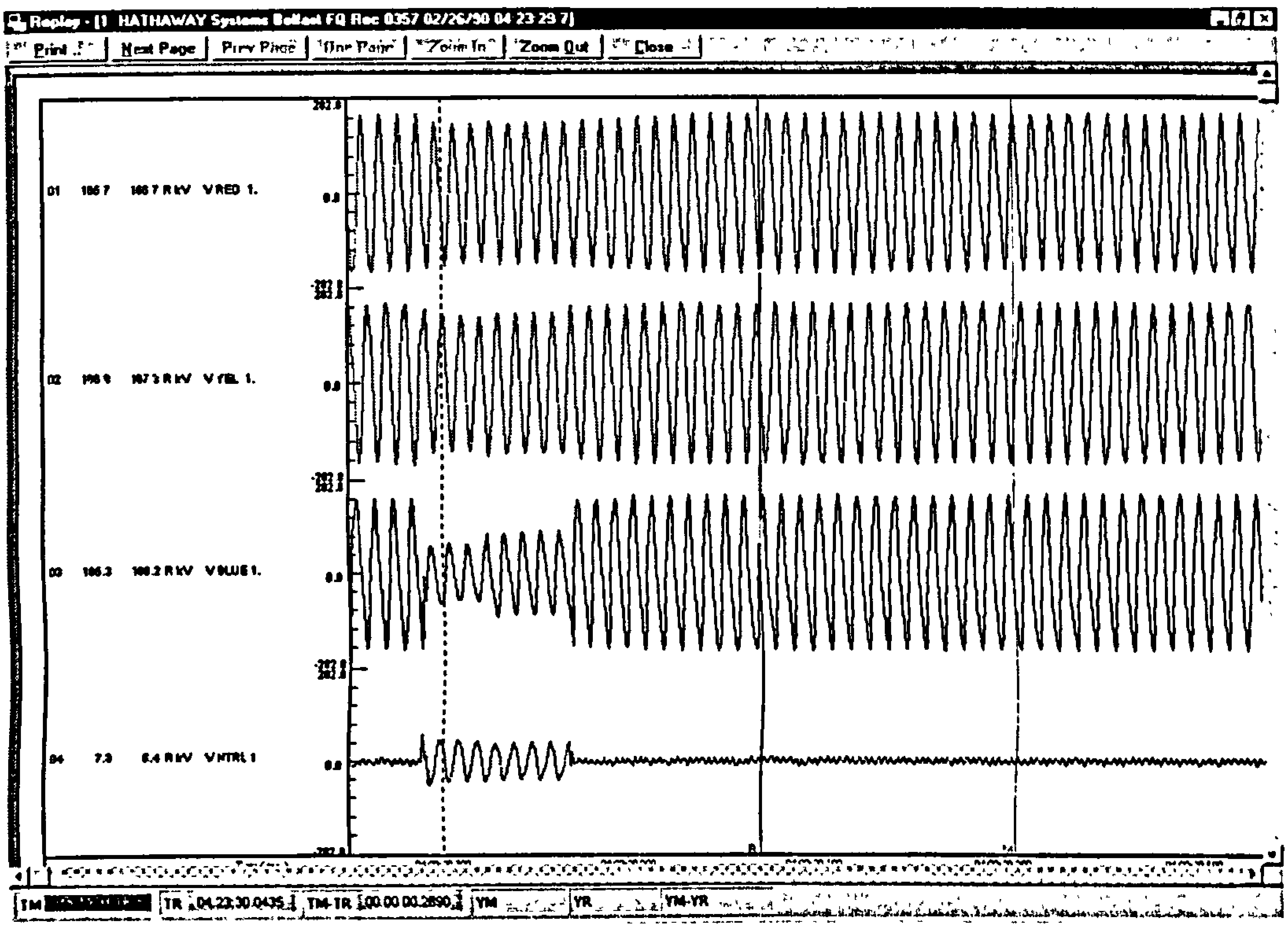

Figure 8.2: A typical blue phase external fault using Hathaway replay software.

Figure 8.2 shows the three phase voltages and the neutral voltage. By inspection it can be seen that the collapse in the blue phase voltage, along with the observation that the other two voltages remain reasonably constant, indicates that there is a blue to ground fault.

Once the data has been converted to ASCII the Hathaway Replay software can't be used so the data is plotted separately by each element. Figure 8.3 shows a blue earth fault. It can be seen that the blue phase voltage collapse's whilst the red and yellow phases remain high but are slightly depressed during the fault. The red, yellow and blue phase currents increase along with the residual current. The spare channels are disconnected.

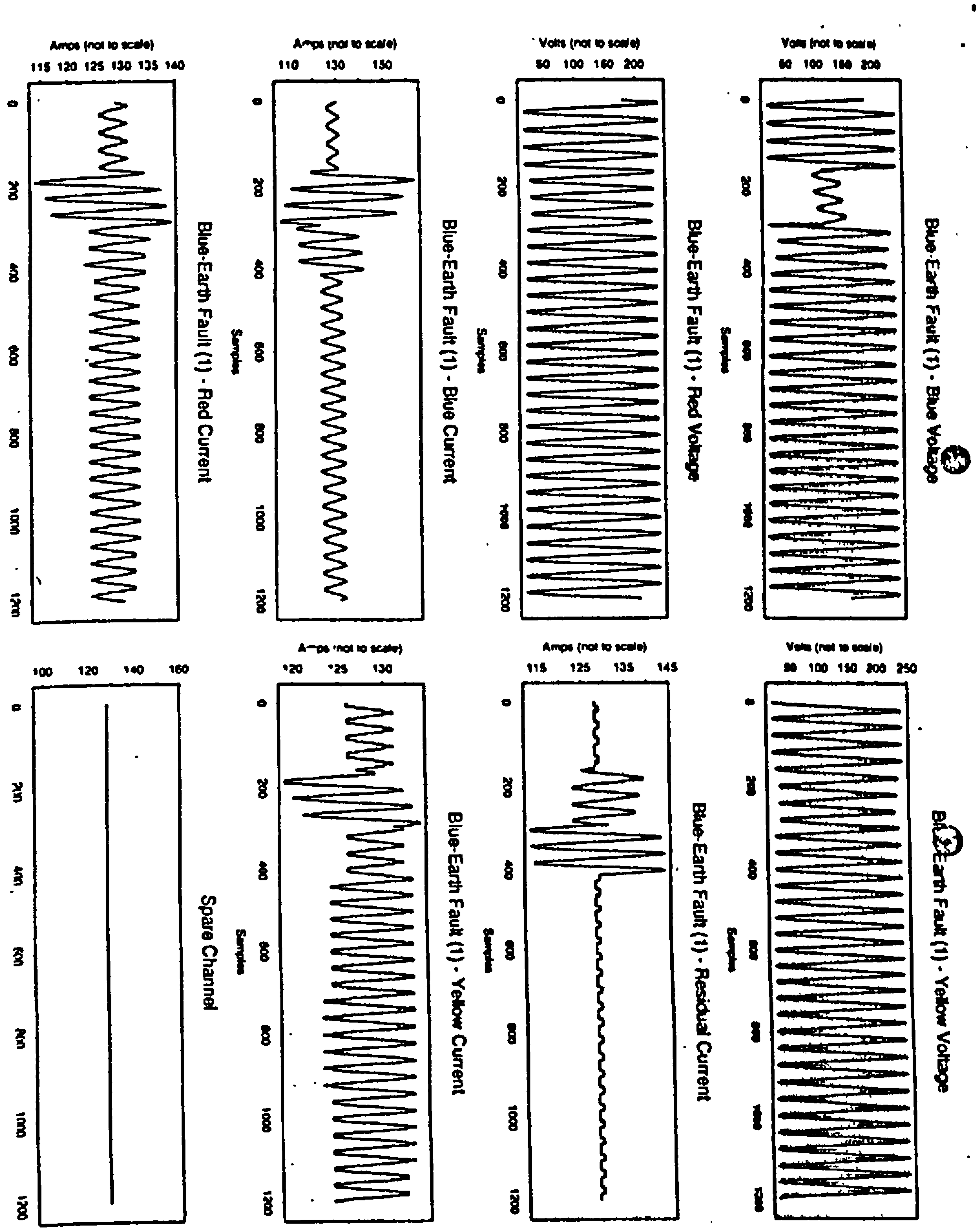

Figure 8.3: Blue earth fault (ASCII data)

Figure 8.4 shows a yellow earth fault. It can be seen that the yellow phase voltage has collapsed but the red and blue voltages remain normal. An increase can be seen for all three phase voltages and the residual current.

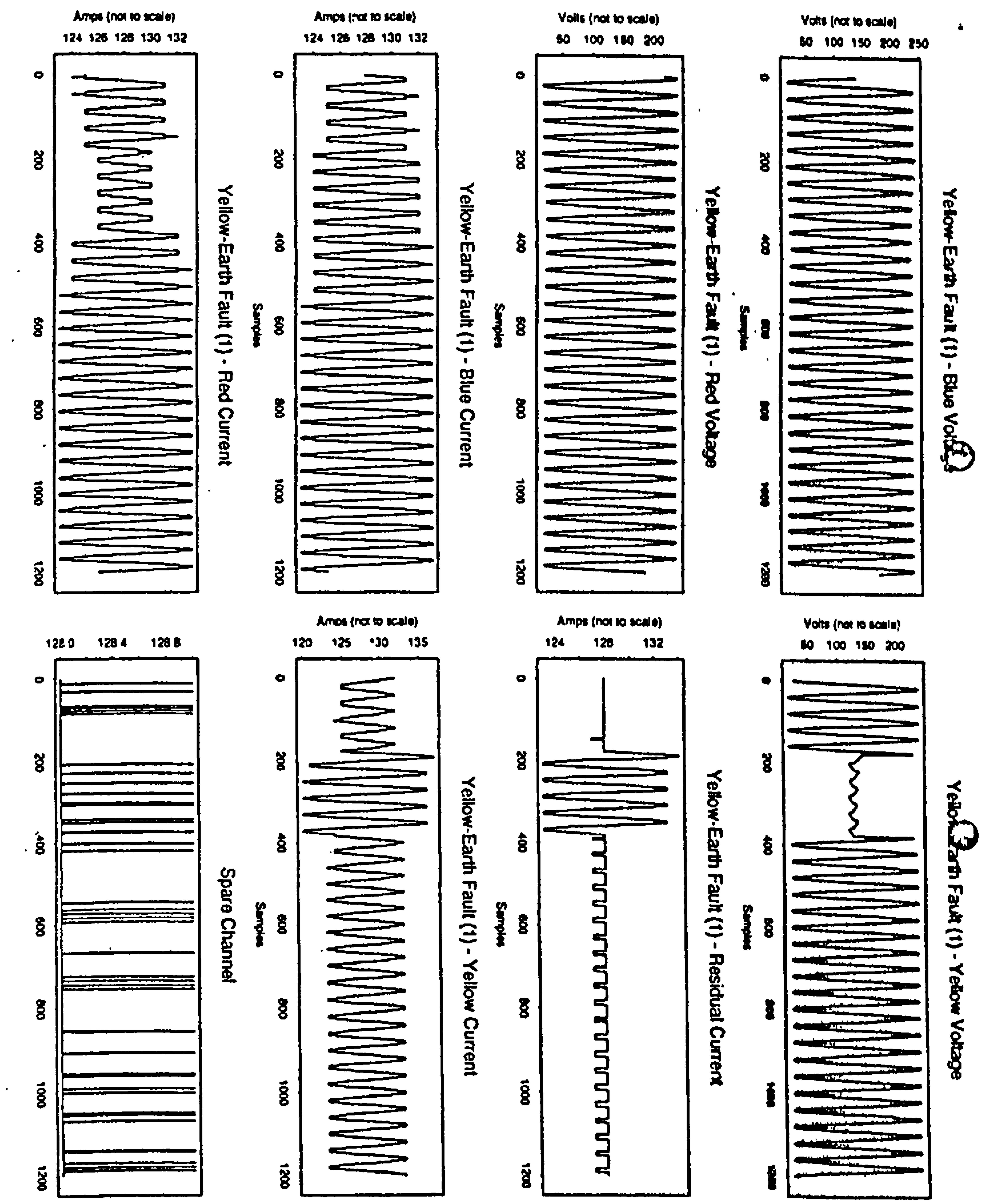

Figure 8.4: Yellow earth fault (ASCII data)

Figure 8.5 shows a three phase fault. It can be seen that all three phase voltages collapse and all three phase currents increase along with the residual current.

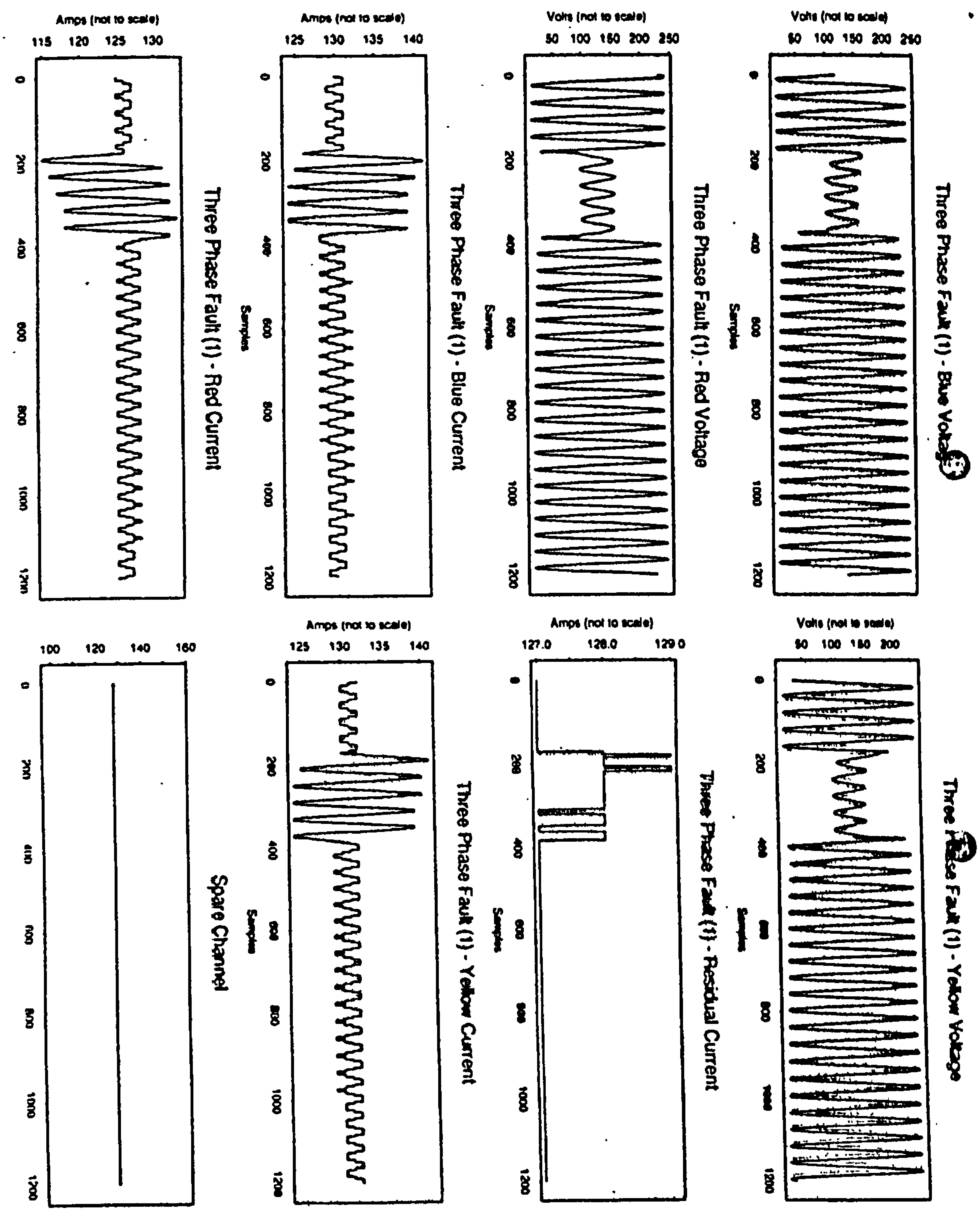

Figure 8.5: Three phase fault (ASCII data)

In the general area of ANN application, it is desirable, from a processing overhead viewpoint, to present the network with a minimal input data vector, whilst preserving the ability of the network to perform its desired task. Thus, where possible, it is important to recognise the explanatory features and variables of the entire input data set. Statistical analysis (e.g. correlation studies etc.) can be useful in extracting the input data features which have most influence on the desired output(s) and are often used in ascertaining the optimal (or minimise) input data vector.

In this particular application, it would have taken a large amount of time and effort to input the fault recorder data to the network in its supplied format. For example, one record, containing the seven analogue parameters of interest, consists of approximately 7 * 1200 ASCII samples (which themselves contain 8 bits of data representing a 0 to 255 decimal value). It is clear that the input of such an example as a single data vector would not be feasible.

However, when the traditional way in which a fault record is analysed is considered a method for reducing the size of this input vector, whilst retaining the information of interest, becomes clear. It is the transitions from pre-fault to during-fault conditions of the various voltage and current levels which ultimately describe the fault class being encountered. Thus, a vector containing this transitional information could be constructed and used to train and test an ANN.

The chosen method of representing these transitions in the data record was to simply use the value of the ratio of "pre" to "during" disturbance rms values for each of the 7 input channels. The derivation of this ratio is a simple task and may be performed via a spreadsheet or other similar means. The presence of post-disturbance voltage was also included (represented via a simple binary code) as input to the SOM in order to ascertain whether the network could discriminate between internal and external disturbances.

For each record , a 2-digit binary code has been used to represent the pre-to during-fault transitions of each of the 7 parameters. A 1 or 0 is used to signify the presence of post-fault voltage. Table 8.2 shows the codes used to indicate fault conditions, where the term significant is used a value would have to be determined by a Power system specialist. For example, where this technique is used in phase comparison unit protection relays for starter operation, a typical value of 12.5% is used for the change detectors.

Table 8.2: Indication of fault conditions

Codes	Events
00	No significant change
10	Significant change
01	Significant decrease

An example of an input vector for a red to earth fault is shown in Table 8.3

Table 8.3: Example of input vector

00	00	01	10	00	00	10	10
Vb	Vy	Vr	$I_{residual}$	Ib	Iy	Ir	$V_{post\ fault}$

Similar vectors were constructed for each of the available records. This process could of course be automated using a computer program. A possible algorithm is shown in Figure 8.6.

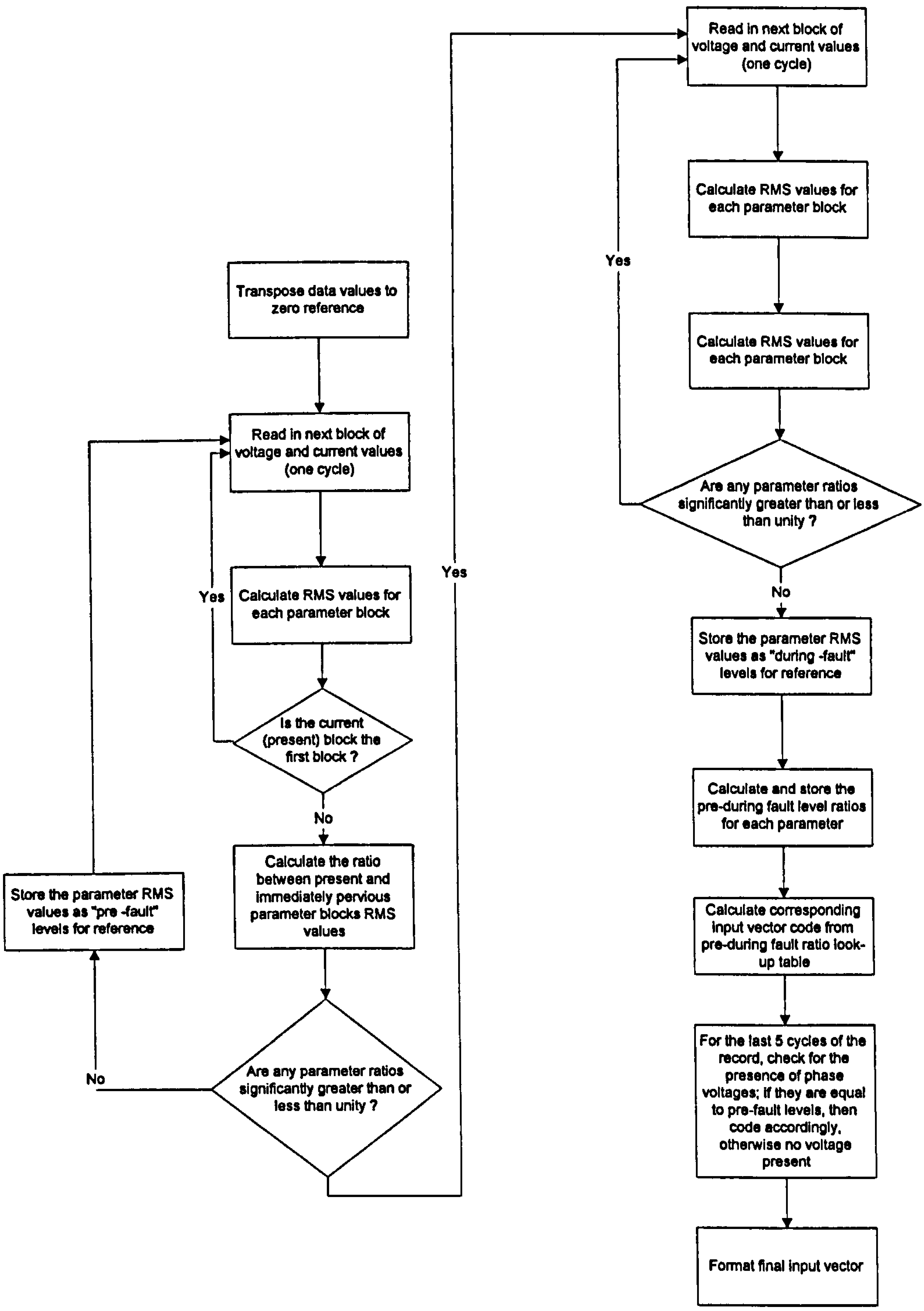

Figure 8.6: Possible pre-processing algorithm for constructing an input vector

8.3.1.2 Results of Experiments to Classify Digital Fault Records

A grey-scale image plot representation has been used, i.e. the darker the colour relating to a node, the greater the relative "winning" frequency of that node.

Three different SOM network topologies were experimented with: a 3*3 map, a 4*4 map and a 5*4 map. It was found that the 5*4 network topology was necessary in order to achieve different responses to *each of the disturbance types.*

Referring to Figure 8.7, the 3*3 network performed quite well in the classification function, displaying a different response to many of the fault classes except for the following, where the network responded in an identical fashion: -

(i) blue-earth phase faults and failed circuit closures

(ii) yellow-red phase and three-phase faults

It would not be possible to classify all types of fault with this particular arrangement.

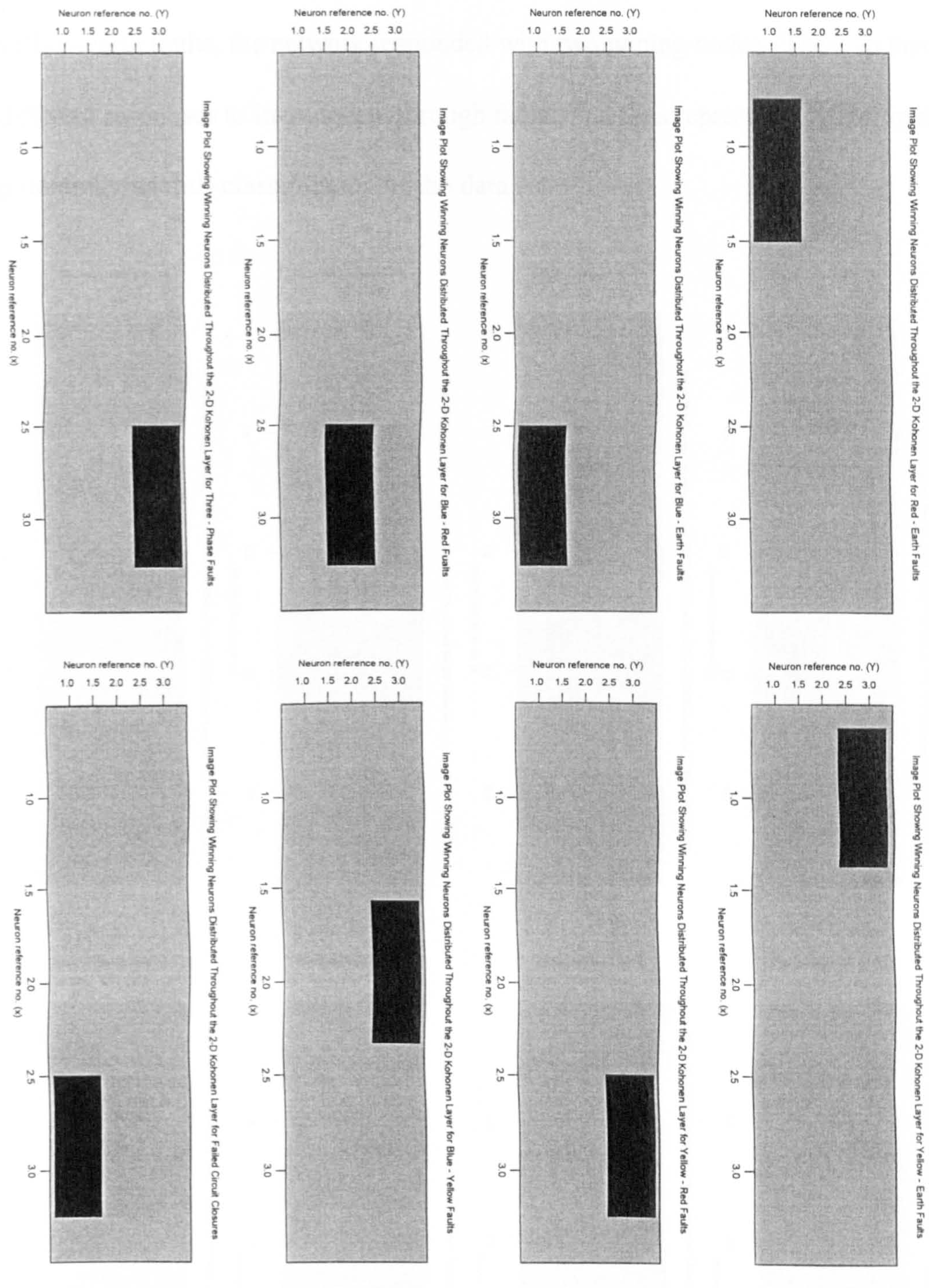

Figure 8.7: Results of the 3*3 SOM network

Referring to figure 8.8, the second network type used, the 4*4 network, responded in a different way to all fault classes except for blue-red phase and three-phase faults, to which the network responded in an identical fashion. Furthermore, for red-earth and yellow-earth faults, the network responded with two wining nodes. These in fact signify different responses to in-zone and through faults, and thus represent a ``finer-grain" (and potentially useful) classification of the data set.

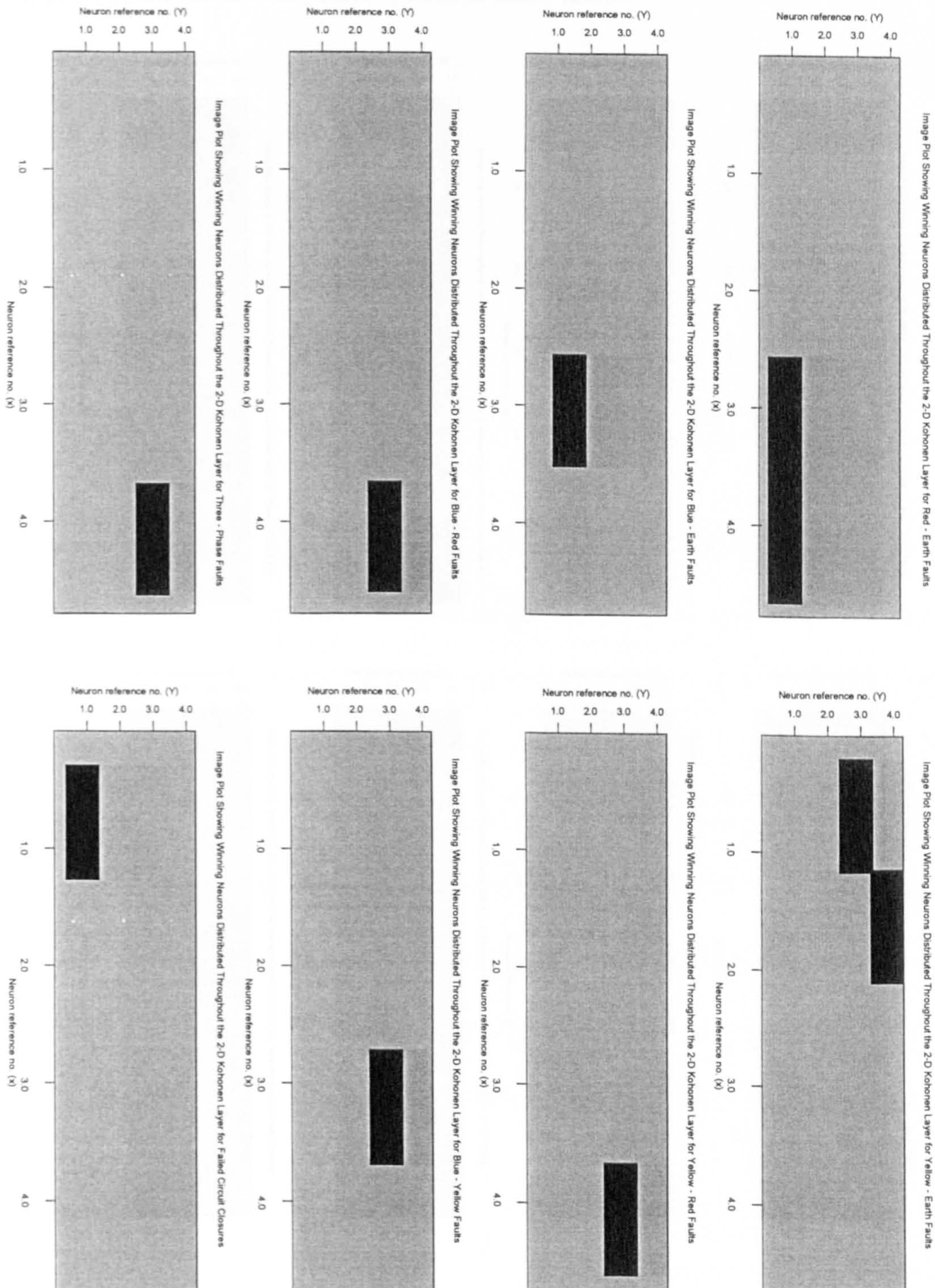

Figure 8.8: Results of the 4*4 SOM network

The final network, containing a 5*4 Kohonen layer shown in Figure 8.9, responds differently to each of the 8 different fault classes with no overlap. Unfortunately, the finer-grain classification of in-zone and through faults was lost for red-phase to earth faults but is retained for yellow-phase to earth faults.

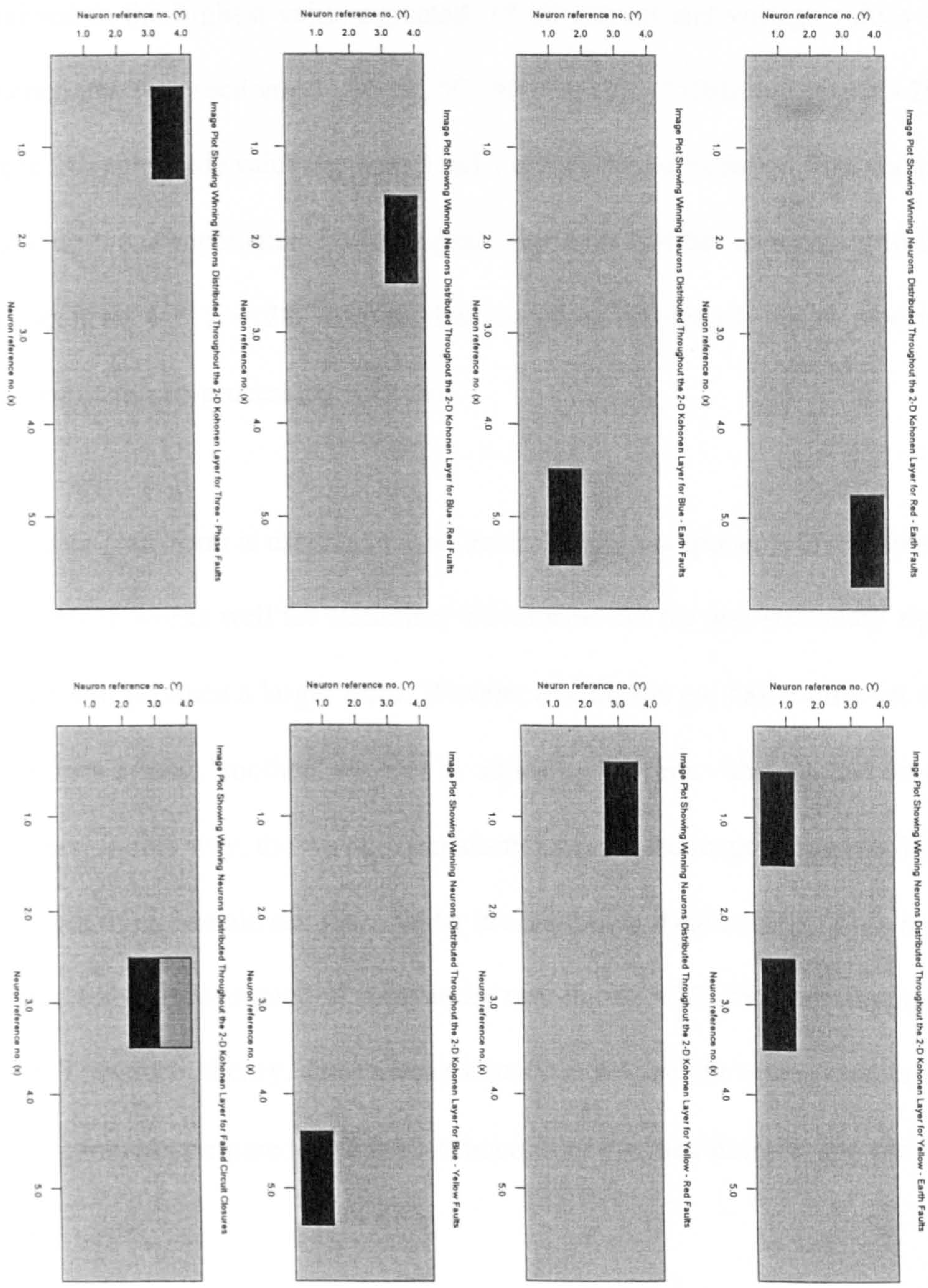

Figure 8.9: Results of the 5*4 SOM network

8.3.1.3 The use of Frequency Components as Input Vectors

An alternative approach for pre-processing the three phase voltages, three phase currents and residual current, is to transform the signals from the time domain into the frequency domain. Each signal, once transformed into the frequency domain could then be normalised to the highest value expected of all current and voltage values and then partitioned into four sections, 0 - 50 Hz, 50 - 100Hz, 100 - 150Hz and 150Hz - 200Hz for example. An amplitude value between 0 and 1 could then be extracted from each partition to make up the components of each signal. The total number of inputs into the SOM would then be 4 * 7 = 28. This method could be translated to an algorithm and implemented as pre-processing software.

The Fourier Transform is used to analyse the frequency components in the time domain. This approach works well for stationary waveforms but for non-stationary signals this approach will produce a larger error. Wavelet transforms are based on a set of signals derived from a basic "mother" wavelet by adjusting the time - shifting and time dilation parameters. In this way, the wavelet transform isolates and magnifies a specific portion of the input time-domain sequence under investigation. Reference [91] has reported an algorithm for feature extraction suitable for training an ANN for fault diagnosis using Wavelet Transform theory. The wavelet transform feature extractor exhibits a superior set of features as compared to feature extractors in the time domain and the frequency domain.

8.3.2 Alternative Neural Network Approach

Subramanian et al [92] reports a different approach to the problem of classifying digital fault records. A three-layer feed-forward model provides an adequate and predictable level of performance. It is characterised by an error back propagation training rule. The training rule is formulated from data sets generated from extensive EMTP simulation tests generated from extensive EMTP simulation tests involving fault type resistance, fault location with phase voltages and currents recorded for each of these situations. The neuron has 6 input layers, which contain the transient data files of fault currents and voltages obtained from fault recorders.

The design of the Neural Network is focused on choosing an appropriate weight matrix model to enable a particular neuron to identify important features of a fault. In the event of a fault recorder producing files pertaining to a phase a to phase b fault, for example, the data is presented to the input layers of all the neurons, but only the neuron assigned to this fault will fire and yield an output 1 while other neurons will be made to give a zero output.

Before the voltage and current data from the fault recorders are fed into the neural network they are pre-processed in two successive sliding data windows of one cycle each to reduce errors, then properly encoded, and finally fed into the input layers of the neurons. 40 samples are used in each window to derive an input vector for three voltages and three currents. Each vector is therefore sized with 240 inputs.

It was found that despite heavy computational requirements due to this method needing sometimes thousands of iterations for a convergence, the feed-forward neural network achieved typical fault classification times of 60 seconds. Which in an off line situation is not excessive. However this does raise concerns that if there were hundreds of fault records following a major storm the analysis time would be excessive. Comparing this with the AFRA software in chapter one a large number of fault records can be analysed in a fraction of a second using a rule base. Subramanian (et al) however go on to explain a hybrid system where a fuzzy set is used to reduce not only the computational burden but also to enhance performance.

8.4 Comparison of ANNs and Rule-Based Techniques

(i) ANNs generate rules through learning by example, if example data is available this can be a quick and easy process.

(ii) Rule-based systems require explicit knowledge in the form of production rules but can be time and effort consuming.

(iii) ANN computing memory is distributed, the connection weights of the ANN are the memory units, and associative - this nature leads to reasonable network response when presented with incomplete, noisy, or previously unseen inputs.

(iv) Rule-based systems make "crisp" decisions, and cannot deal with situations not described explicitly within the rule base.

(v) ANNs cannot efficiently describe why a particular output was produced and are often thought of as a "black-box".

(vi) Rule-based systems can effectively explain their output through providing a step-by-step description of their rule-based decision process, if designed with this facility as a requirement.

It is clear that ANNs and rule-based systems should be viewed as complementary, as opposed to competitive. The application of ANNs as described in this paper could complement the existing rule-based system, producing a more comprehensive overall package, which could deal with more disturbance types than at present.

8.5 Conclusions

This chapter has demonstrated a successful application of SOM ANNs and feed-forward Kohonan ANNs to the classification of digital fault record data by primary system fault class. The completed work is preliminary in nature and an overview of an extension to this work, involving the extraction of frequency components from the digital fault record data and using these as input to a SOM network, has been described.

Future work should concentrate on the development of software to automatically produce an input data vector from a fault record and investigation should also be carried out into possible refinements of such techniques to provide output information relating to, for example, the location of the fault. However, this would require such information to be available during the train/test procedure.

Chapter IX

CONCLUSION

This research has shown that the application of Artificial Intelligence techniques in Power System analysis is of benefit and improves the capability of engineers. Deregulation of electricity markets will bring with it more complex trading arrangements between power producers, transmission operators and their customers.

The use of software tools is now an integral part of the application of protection in complex situations and in the investigation of unwanted operation or failure to operate of protection systems. It provides a more accurate and efficient way of carrying out these investigations. The implementation of the Automatic Fault Record Analysis (AFRA) software will reduce the amount of effort required by a protection engineer when analysing a large number of fault records.

The development of expert systems is a continuous process as new knowledge is gained in the field of artificial intelligence and new expert system development tools are built. Efforts are being made for on-line application of expert systems in ECC as preventive control under normal/alert conditions and as a corrective control during a disturbance. This will enable a more secure power system operation. Considerable scope exists in the development of expert systems and their application to power system operation and control.

There are many types of Artificial Neural Networks available along with a number of techniques used for their implementation. Although the mathematical concepts are not new, many of them were recorded more than fifty years ago; the introduction of fast computers has enabled many of these concepts to be used for today's complex problems.

A review of the Evolutionary Computing family of optimisation methods with particular emphasis on the Genetic Algorithm, which is used in investigation within this thesis has shown the Genetic Algorithm to be a useful tool in designing Neural Networks. Various other popular methods of optimisation were then reviewed.

The forecasting model in chapter six has been used to produce a forecast of the load in the 24 hours of the forecast day concerned, using data provided by an Italian power company[93]. The results obtained are promising. In this particular case, the comparison between the results from the GA_ANN and BP_ANN shows that the GA_ANN does not provide a faster solution than the BP_ANN. This could be due to the fact that the initial randomly selected starting point is a poor one. The size of the problem is very large and as such the amount of memory and computation time is large too. This points to the direction of parallel processing techniques being integrated with evolutionary computing to solve complex practical problems.

An Evolutionary Programming approach has been developed for solving the Fault Section Estimation problem including malfunctions of protection relays and/or circuit breakers and multiple fault cases. A comparison is made with the GA approach at the same time. Two different population sizes are tested for each case. In general, EP showed faster computational speed than GA with an average factor of 13 times more. The final results were almost the same. The convergence speed (the required number of generations to get an optimum result) is a very important factor in real time applications. Test results show that EP is better than GA. However, as both EP and GA are evolutionary algorithms, their efficiencies are largely dependent on the complexity of the problem, which might differ from case to case. EP is also ideal for parallel processing computer systems or hardware. Therefore, with this kind of equipment such as transputers, it is possible to solve FSE problem faster and with high efficiency.

A successful application of SOM ANNs and feed-forward Kohonan ANNs has been demonstrated for the classification of digital fault record data by primary system fault class. The completed work is preliminary in nature and an overview of an extension to this work, involving the extraction of frequency components from the digital fault record data and using these as input to a SOM network, has been described.

Future work should concentrate on the development of software to automatically produce an input data vector from a fault record and investigation should also be carried out into possible refinements of such techniques to provide output information relating to, for example, the location of the fault. However, this would require such information to be available during the train/test procedure.

Work continues in the application of Artificial Intelligence techniques in Power System analysis. Short term demand forecasting system marginal price predictions and analysis of transformer gas discharges are examples of current areas of research and development in the UK. Development of tools in this area is already proving to be beneficial to engineers. Although slower than other sectors Artificial Intelligence techniques will, in future, play an important role in Power System analysis and management.

REFERENCES

[1] ZR20 - Computer Program for Evaluation and Application of Distance Relays - User's Guide, Power Systems Management Corporation, Pennsylvania, USA.

[2] FR10 - Computer Program for Evaluation of Transmission Protection Performance using information from Fault Recorders - User Guide, Power Systems Management Corporation, Pennsylvania, USA,

[3] C.L Coull, J. R. Mcdonald, M. Turnay, .1. R. Smith, J. Goody, B. Gwyn -Dynamic Modelling of Protection Performance - UPEC 1994, page 215.

[4] J L H Goody A M Carter, L P Cavero, & J Hawker - Automatic Evaluation of Transmission Protection Using Information From Digital Fault Recorders – IEE DPSP 1997 IEE Conference Publication 434, page 414.

[5] F F Wu, 'Real time network security monitoring, assessment and optimization', International Journal of Electrical Power and Energy Systems, 'Vol 10, No 2, 1988, 83-100.

[6] E Handschin and A Petroianu, Energy Management Systems - Operation and Control of Electric Energy Transmission Systems, Springer-Verlag, Germany, 1991.

[7] Henry & Mishkoff, Understanding Artificial Intelligence, BPB Publications, New Delhi, 1986.

[8] Dan W Petterson, Introduction to Artificial Intelligence and Expert Systems, Prentice - Hall, New Delhi, 1992.

[9] Dennis Merritt, Building Expert Systems in Prolog, Springer-Verlag, New York, 1986.

[10] James Martin and Steven Oxman, Building Expert Systems - A Tutorial, Prentice-Hall, New Jersey, 1988.

[11] Neal J Balu, Rambbabu Adapa, Gerry Cauley, Mark Lauby and Dominic J Maratukulam, 'Review of expert systems in bulk power planning and operation', Proceedings of the IEEE, 80(5), 1992, 727-731.

[12] Keinosuke Matsumoto, Toshiaki Sakaguchi, Richard J Kafka and Mike M Adibi, 'Knowledge based systems as operational aids in power system restoration', Proceedings of the IEEE, Vol 80, No 5, 1992, 689-697.

[13] Z Z Zhang, G S Hope and O P Malik, 'Expert systems in electric power systems - a bibliographical survey', IEEE Transactions on Power Systems, Vol 4, No 4, 1989, 1335-1362.

[14] C C Liu and T Dillon, State -of-the-art of expert system applications to power systems, 'International Journal of Electrical Power and Energy Systems, Vol 14, No 2, 1992, 86-96

[15] CIGRE Task Force. Survey on Expert Systems in Alarm Handling. Electra, 1991.

[16] E Cardozo and S N Taludar, 'A distributed expert system for fault diagnosis', IEEE Transactions on Power Systems, Vol 3, No 2, 1988, 641-646.

[17] K Komai, T Sakaguchi and S Takeda, 'Power system fault diagnosius with an expert system enhanced by the general problem solving method', Proceedings of the IASTED High Technology in Power Industry, 1988.

[18] C Fukui and J Kawakami, 'An expert system for fault section section using information from protective relays and circuit breakers', IEEE Transactions on Power Delivery, Vol 1, No 4, 83-87, 1986.

[19] K Tomsovic, C C Liu, P Ackerman and S Pope, 'An expert system as a dispatch bid for isolation of line section faults', IEEE Transactions on Power Delivery, Vol 2, No 3, 1987, 736-743.

[20] Yasuji Sekine, Yoshiakira Akimoto, Masahiko Kunugi, Chihiro Fukui and Shinta Fukui, 'Fault diagnosis of power systems', Proceedings of the IEEE, Vol 80, No 5, 1992, 6743-683.

[21] T Sakaguchi and K Matsumoto, 'Development of a knowledge based system for power system restoration', IEEE Transactions on Power Apparatus and Systems, Vol 102, No 2, 1983, 320-329.

[22] Y Kojima, S Warashina, S Nakamura and K Matsumoto, 'Development of a guidance method for power system restoration', IEEE Transactions on Power Systems, Vol 4, No 3, 1989, 1219-1227.

[23] Y Kojima, S Warashina, M Kato and H Watanabe, 'The development of power system restoration for a bulk power system by applying knowledge engineering techniques', IEEE Transactions on Power Systems, Vol 4, No 3, 1989, 1228-1235.

[24] Model-based analysis of protection system performance,
Bell, S.C., McArthur, S.D.J, McDonald, J.R., Burt, G.M., Mather, R.M. and Cumming, T., IEE Proceedings - Generation, Transmission and Distribution, Vol 145, No. 5, Sept 1998, 547-552.

[25] M El-Sharkawi and D Niebur (editors), Artificial neural networks with applications to power systems, IEEE Tutorial course, IEEE Catalog Number 96TP112-0, 1996.

[26] Loi Lei Lai, Intelligent Systems Applications in *Power Engineering:* Evolutionary Programming and Neural Networks, John Wiley & Sons Ltd, 1998.

[27] A T P So and W L Chan, Intelligent Building Systems, Kluwer Academic Publishers, Norwell, 1999.

[28] W S McCulloch and W H Pitts, 'A logical calculus of the ideas immanent in nervous activity', Bulletin of Mathematical Biophysics, Vol 5, 1943, 115-133.

[29] F Rosenblatt, Principle of Neurodynamics: Perceptrons and the Theory of Brian Mechanisms, Spartan, New York, 1962.

[30] S Grossberg, 'Some nonlinear networks capable of learning a spatial pattern of arbitrary complexity', Proceedings of the National Academy of Sciences, USA, Vol 59, 1968, 368-372.

[31] G A Carpenter and S Grosssberg, 'A massively parallel architecture for a self-organising neural pattern recognition machine', Computer Vision, Graphics and Image Processing, Vol 37, 1987, 54-115.

[32] T Kohonen, 'Self-organized formation of topologically correct feature maps', Biological Cybernetics, Vol 43, 1982, 59-69.

[33] T Kohonen, Self-organising Maps, Springer-Verlag, New York, 1995.

[34] Wilson, S.W.: "Classifier Systems and the Animate Problem", Research memo RIS-27r, Rowland Institute for Science, Cambridge, 1986.

[35] Maudlin, M.L.: "Maintaining Diversity in Genetic Search", in *AAAI Proceedings, National Conference on Artificial Intelligence*, 1984, pp. 247-250.

[36] Fogel, D.B.: "An Introduction to Evolutionary Computation", in *Australian Journal of Intelligent Information Processing Systems*, Vol 1, No. 2, June 1994, pp. 34-42.

[37] De Jong, K.A. and Spears, W.M.: "An Analysis of the Interactive Roles of Population Size and Crossover in Genetic Algorithms", in *Proceedings of the First*

Workshop of Parallel Problems Solving from Nature, Springer-Verlag, Berlin, 1990, pp. 38-47.

[38] Grefenstette, J.J.: "Optimisation of Control Parameters for Genetic Algorithms", in *IEEE Transactions, Systems, Man and Cybernetics*, Vol. SMC-16, No. 1, Jan/Feb 1986, pp. 122-128.

[39] Wong, K.P. and Wong, Y.M.: "Development of Parallel Geetic Algorithms" in *Australian Journal of Intelligent Information Processing Systems*, 1994, Vol. 1(1), pp. 51-57.

[40] Cohoon, J.P., Hedge, S.U., Martin, W.N. and Richards, D.S.: "Distributed Genetic Algorithms for the Floor Plan Design Problem", in *IEEE Transactions on Computer-Aided Design*, Vol 10(4), 1991, pp. 483-492.

[41] Cohoon, J.P. and Paris, W.D.: "Genetic Placement", *IEEE Transactions on Computer-Aided Design*, Vol 6(6), pp. 956-964.

[42] Bala, J.W. and De Jong, K.: "Generation of Feature Detectors for Texture Discrimination by Genetic Search", in *Proceedings of the Second International IEEE Conference on Tools for AI*, 1990, pp. 812-818.

[43] Mandava, V.R., Fitzpatrick, M. and Piclens, D.R.: "Adaptive Search Space Scaling in Digital Image Restoration", in *IEEE Transactions on Medical Imaging*, 1989, 8(3), pp. 251-262.

[44] Dontas, K. and De Jong, K.: "Discovery of Maximal Distance Codes Using Genetic Algorithms", in *Proceedings of the Second International IEEE Conference on Tools for AI*, 1990, pp. 805-811.

[45] Parker, J.K. and Goldberg, D.E.: "inverse Kinematics of Redundant Robots Using Genetic Algorithms", in *IEEE International Conference on Robotics and Automation*, 1989. Pp. 271-276.

[46] Angeline, P.J.: "Adaptive and Self-Adaptive Evolutionary Computation", in *Computational Intelligence – A Dynamic System Perspective*, IEEE Press, 1995, pp. 152-163.

[47] Walter, D.C. and Sheble, G.B.: "Genetic Algorithm Solution for Economic Dispatch with Valve Point Loading", in *IEEE PES Summer Meeting*, 1992, Seattle, Paper Number 92 SM 414-3 PWRS.

[48] Yin, X. and Germany, N.: "Investigations on Solving the Load Flow Problem by Genetic Algorithms", in *Electr. Power Sys. Res.*, 1991, 22, pp. 151-163.

[49] Wong, K.P., and Li, A.: "Solving the Load-Flow Problem using Geneteic Algorithm", in *Proceedings of IEEE International Conference on Evolutionary Computation*, Vol. 1, 1995, pp. 103-108.

[50] Dasgupta, D and McGregor, D.R.: "Thermal *Unit Commitment* Using Genetic Algorithms", in *IEE Proceedings*, Part C, 1994, 141, (5), pp. 459-465.

[51] Kazarlis, S.A., Bakirtzis, A.G. and Petridis, Z.V.: "A Genetic Algorithm Solution to the Unit Commitment Problem", in *IEEE PES Winter Meeting*, 1995, New York, Paper Number 95 WM 152-9 PWRS.

[52] Beckwith, S.F. and Wong K.P.: "A Genetic Algorithm Approach for Electric Pump Scheduling in Water Supply Systems", *in Proceedings of the International Conference on Evolutionary Computation*, December 1995, Volume One, pp. 21-26.

[53] Wong, K.P., Wong, Y.W.: "Floating-Point Number Coding Method for Genetic Algorithms", in *Proceedings of the 1993 Australian and New Zealand Conference on Intelligent Systems*, IEEE, 1993, pp. 512-516.

[54] Heitkoetter, J. and Beasley, D. ads. (1996) "the Hitch-Hiker's Guide to Evolutionary Computation: A List of Frequently Asked Questions (FAQ)", *USENET: comp.ai.genetic. Available via anonymous FTP from rtfm.mit.edu:/pub/usenet/news.amswers/ai-faq/genetic/.*

[55] Fogel, L.J.: "Autonomuos Automata", in *Industrial Research*, Vol. 4, 1962, pp. 14-19.

[56] Rechenberg, I.: "Cybernetic Solution Path for an Experimental Problem", in *Royal Aircraft Establishment, Lib. Trans*, 1122, Farnborough, 1965.

[57] Rechenberg, I.: "Evolutinsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biolgischen Evolution", Stuttgart: Fromman-Holzboog Verlag, 1973.

[58] Schwefel, H.P.: "Kibernetische Evolution als Strategie der Experimentellen Forschung in der Stromungstechnk", Diploma thesis, Tech. Univ. of Berlin, 1965.

[59] Schwefel, H.P.: "Numerical Optimisation of Computer Models", Chichester, UK.: John Wiley, 1981.

[60] Schwefel, H.P.: "Collective Phenomena in evolutionary Systems", *31st Annual Metting of the International Society for General Systems Research*, Budapest, 1987, pp. 1025-1033.

[61] Bartholomew-Biggs, M.C.: "Algorithms for General Constrained Non-linear Optimisation", in *Algorithms for Continuos Optimisation. The State of the Art*, Spedicato, ed., Kluwer, Netherlands, 1994, pp. 167-207.

[62] Wong, K.P. and Wong, Y.W.: "Short Term Hydrothermal Scheduling. Part 1: Simulated Annealing Approach", in *IEE Proceedings, Generation, Transmission and Distribution*, Vol. 141, No. 5, September 1994.

[63] Bellman, R.E.: "Project Rand: Dynamic Programming of Continuos Processes", Santa Monica, California, Rand Corporation, 1954.

[64] Cormen, T.H., Leiserson, C.E., Rivest, R.L.: "Introduction to Algorithms", MIT Press, Cambridge, Massachusetts, 1992.

[65] Smith, D.K.: "Dynamic Programming. A Practical Introduction", Ellis Horwood, England, 1991.

[66] Aarts, E and Korst, J.M.: "*Simulated Annealing and Boltzmann Machines: A Stochastic Approach Combinatorial Optimisation and Neural Computing*", John Wiley, New York, 1989.

[67] Wong, K.P. and Fung, C.C.: "Simulated Annealing based Economic Dispatch Algorithm", in *IEE Proceedings-C*, Vol. 140, No. 6, November 1993, pp.509-515.

[68] Rubistein, R.Y.: "*Simulation and Monte Carlo Method*", John Wiley, New York, 1981.

[69] Kirkpatrick, S., Gelatt, C. and Vecchi, M.P.: "Optimisation by Simulated Annealing", in *Science*, 1983, 220, (4598), pp. 671-680.

[70] Styblinski, M.A. and Tang, T.S.: "Experiments in Non-convex Optimisation: Stochastic Approximation with Function Smoothing and Simulated Annealing", in *Neural Networks*, 1990, 3, pp. 467-483.

[71] Azencott, R. ed.: "*Simulated Annealing: Parallelisation Techniques*", John Wiley, united States, 1992.

[72] K. P. Wong and C.C. Fung 'Simulated annealing based economic dispatch algorithm' , IEE Proc. Part C, 1993, vol 6, 509 – 515.

[73] G Gross and F D Galiana, 'Short term load forecasting', Proc. of IEEE, Vol. 75, No. 12, 1987, pp. 1558-1573.

[74] M T Hagan and S M Behr, 'The time series approach to short term load forecasting', IEEE Transactions on Power Systems, Vol. 2, No. 3, 1987, pp. 785 - 791.

[75] A D Papalexopoulos and T C Hesterberg, 'A regression-based approach to short-term load forecasting', IEEE Transactions on Power Systems, Vol. 5, No. 4, 1990, pp. 1535-1547.

[76] S Rahman and R Bhantnagar, 'An expert system based algorithm for short-term load forecast', IEEE Transactions on Power Systems, Vol. 3, No. 2, 1988, pp. 392-399.

[77] A S Dhdashti, J R Tudor and M C Smith, 'Forecasting of hourly load by pattern recognition: a deterministic approach', IEEE Transactions on Power Apparatus and Systems, Vol. 101, 1982, pp. 900-910.

[78] M Caciotta, R Lamedica, V Orsolini Cencelli, A Prudenzi and M Sforna, 'Application of artificial neural networks to historical data analysis for short-term

electric load forecasting', European Transactions on Electrical Power, Vol. 7, 1997, pp. 49-56.

[79] T Maifield and G Sheble, 'Short term load forecasting by neural network and a refined genetic algorithm', Electrical Power Systems Research, Vol. 31, pp. 9-14.

[80] L L Lai, A G Sichanie, N Rajkumar, E Styvaktakis, M Sforna and M Caciotta, 'Practical application of object oriented techniques to designing neural networks for short-term electric load forecasting', Proceedings of the Energy Management and Power Delivery Conference, IEEE Catalogue No 98EX137, March 1998, pp. 559-563.

[81] E T H Heng, D Srinivasan and A C Liew, 'Short term load forecasting using genetic algorithm and neural networks', Proceedings of the Energy Management and Power Delivery Conference, EEE Catalogue No 98EX137, March 1998, pp. 576-581.

[82] L L Lai, Intelligent system applications in power engineering - evolutionary programming and neural networks, John Wiley & Sons, 1998.

[83] D Srinivasan, F Wen, C S Chan and A C Liew, 'A survey of applications of evolutionary computing to power system', Proceedings of the International Conference on Intelligent Systems Applications to Power Systems, Editors: O A Mohammed & K Tomsovic, IEEE Catalog Number 96 TH 8152, Jan/Feb 1996, 35-41.

[84] V Miranda, D Srinivasan and L M Proenca, 'Evolutionary computation in power systems', Proceedings of the 12th Power Systems Computation Conference, PSCC, Germany, Aug 1996, 25-40.

[85] C Fukui and J Kawakami, 'An expert system for fault section estimation using information from protection relays and circuit breakers', IEEE Transactions on Power Delivery, Vol.1, No. 4, Oct 1986, 83-90.

[86] L L Lai, F Ndeh-Che, K H Chu, P Rajroop and X F Wang, 'Design of neural networks with genetic algorithms for fault section estimation', Proceedings of the 29th Universities Power Engineering Conference, Galway, Ireland, Vol 2, 1994, 596-599.

[87] H T Yang, W Y Chang, and C L Huang, 'A new neural networks approach to on-line fault section estimation using information of protection relays and circuit breakers', IEEE Transactions on Power Delivery, Vol. 9, No. 1, January 1994, 220-230

[88] F Wen and Z Han, 'An Evolutionary optimisation method to fault section estimation using information from protection relays and circuit breakers', Proceedings of the International Conference on Power System Technology, Beijing, China, International Academic Publishers, Oct 1994, 1051-1055.

[89] J T Ma and L L Lai, 'Evolutionary programming approach to reactive power planning', IEE Proceedings, Gener. Transm. Distrib., Vol 143, No 4, July 1996, 365-370.

[90] J Momoh and D T Rizy , Application of wavelet theory to power distribution systems for fault detection, Proceedings of the International Conference on Intelligent Systems Application to Power Systems, IEEE Catalogue No 96Th8152, Jan/Feb, 345-350.

[91] J Downes, J Goody and B Gwyn - The Use of Software Tools for Investigating the Performance of Power System Protection, CIGRE - study committee 34 colloquium and meeting 11 - 17 June 1995, Stockholm.

[92] R Subramanian, A. S. Mereikhi, A. Marashi, A Samuel, A.H. Al-Baya, C. A. De Couto, N. N. Puri, Expert System for Automatic Performance Monitoring of Utility Power Network, CIGRE, Session 34-101, 1998.

[93] A Neural Networks: A Technology Handbook for NeuralWorks Professional II/PLUS@, NeuralWare, 1993.